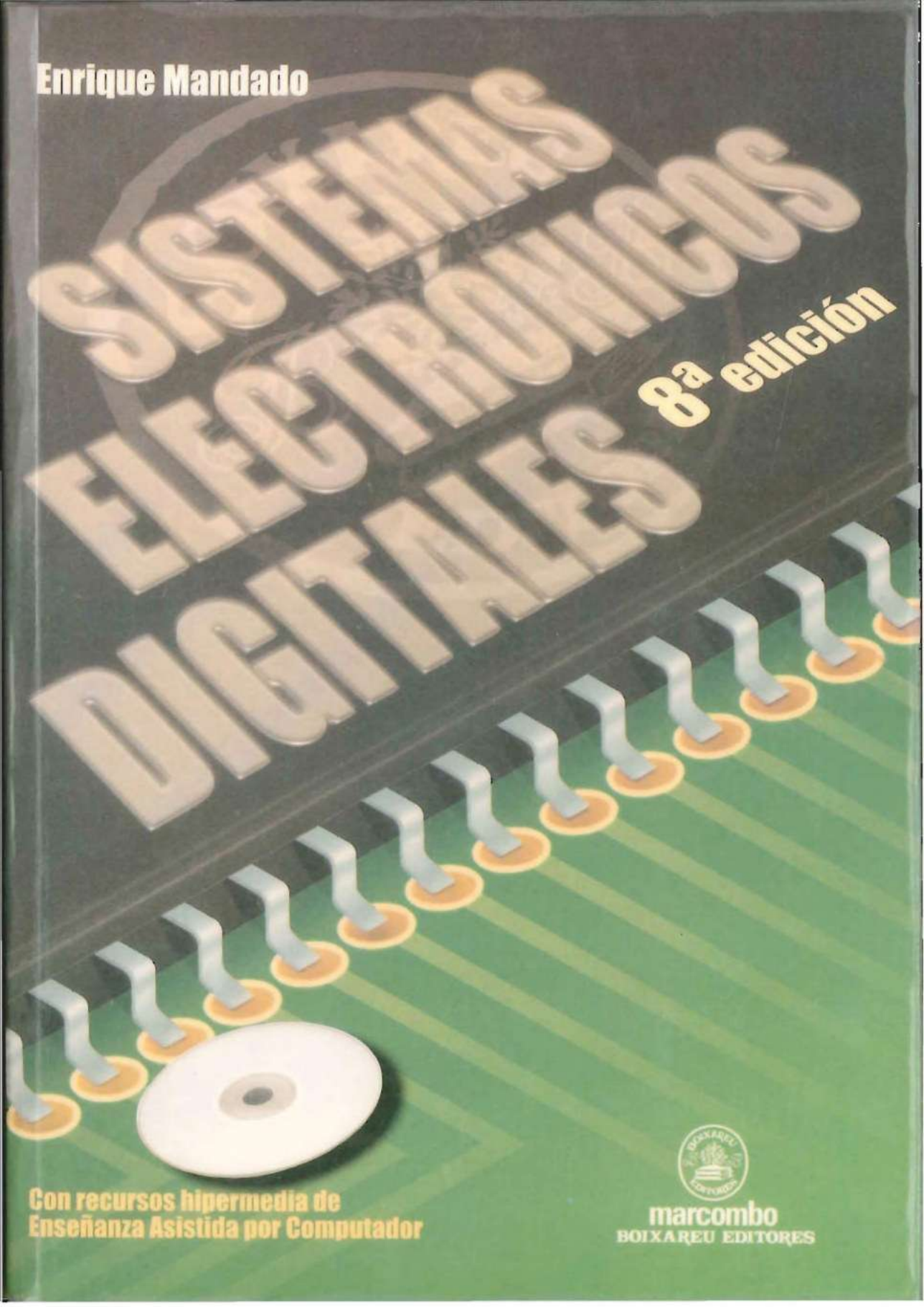
Enrique Mandado

SISTEMAS ELECTRÓNICOS DIGITALES

8ª edición



Con recursos hipermedia de Enseñanza Asistida por Computador

marcombo
BOIXAREU EDITORES

SISTEMAS ELECTRONICOS DIGITALES

Tomo I
Circuitos combinacionales y secuenciales

Amigo lector:

La obra que usted tiene en sus manos posee un gran valor. En ella, su autor, ha vertido conocimientos, experiencia y mucho trabajo. El editor ha procurado una presentación digna de su contenido y está poniendo todo su empeño y recursos para que sea ampliamente difundida, a través de su red de comercialización.

Usted puede obtener fotocopias de las páginas del libro para su uso personal. Pero desconfíe y rehúse cualquier ejemplar "pirata" o fotocopia ilegal del mismo porque, de lo contrario, contribuiría al lucro de quienes, consciente o inconscientemente, se aprovechan ilegítimamente del esfuerzo del autor y del editor.

La reprografía indiscriminada y la piratería editorial, no solamente son prácticas ilegales, sino que atentan contra la creatividad y contra la difusión de la cultura.

PROMUEVA LA CREATIVIDAD
RESPETE EL DERECHO DE AUTOR

ENRIQUE MANDADO
CATEDRATICO DE TECNOLOGIA ELECTRONICA
DEPARTAMENTO DE TECNOLOGIA ELECTRONICA
DE LAS ESCUELAS TECNICAS DE INGENIERIA
DE LA UNIVERSIDAD DE VIGO

SISTEMAS ELECTRONICOS DIGITALES

Tomo I
Circuitos combinacionales y secuenciales

marcombo
BOIXAREU EDITORES

BARCELONA

Copyright © 1998 por E. Mandado
Reservados todos los derechos
de publicación, reproducción, préstamo,
alquiler o cualquier otra forma de cesión del
uso de este ejemplar en cualquier idioma por
MARCOMBO, S.A.
Gran Via de les Corts Catalanes, 594
08007 Barcelona (España)

8.ª edición

Quedan rigurosamente prohibidas, sin la autorización escrita de los titulares del "Copyright", bajo las sanciones establecidas en las leyes, la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, y la distribución de ejemplares de ella mediante alquier o préstamo públicos, así como la exportación e importación de esos ejemplares para su distribución en venta, fuera del ámbito de la Comunidad Económica Europea.

ISBN 84-267-1169-3 (obra completa)
ISBN 84-267-1170-7 (tomo 1)
Depósito legal: B-39656-1998
Impreso en España
Printed in Spain
Fotocomposición: ApG - Entenza, 218 - 08029 Barcelona
Impresión: Gràfiques 92, S.A. - Avda. Can Sucarrats, 91-08191 Rubí

Indice general

PROLOGO A LA OCTAVA EDICION XI
INTRODUCCION A LA OCTAVA EDICION XIII

CAPITULO 1. Sistemas y códigos de numeración 1

1.1 Generalidades 1
1.2 Representación de los números. Sistemas de numeración 1
1.3 Sistema binario 2
1.4 Sistema octal 5
1.5 Sistema hexadecimal 6
1.6 Códigos binarios 7
1.6.1 Códigos decimales codificados en binario 7
1.6.2 Códigos binarios continuos y cíclicos 10
1.7 Códigos alfanuméricos 13
1.8 Códigos detectores de errores 14
1.9 Códigos correctores de errores 17

CAPITULO 2. Algebra de Boole 21

2.1 Definición y postulados 21
2.2 Teoremas del álgebra de Boole 22
2.3 Función de un álgebra de Boole 29
2.4 Tabla de verdad de una función lógica 33
2.5 Funciones importantes de un álgebra de Boole 34
2.5.1 Función O-exclusiva 34
2.5.2 Función equivalencia o comparación 36

CAPITULO 3. Sistemas combinacionales 39

3.1 Generalidades 39
3.2 Simplificación de las funciones lógicas 40
3.2.1 Métodos tabulares de Karnaugh y Veitch de simplificación de las funciones lógicas 42
3.2.2 Método numérico de Quine-McCluskey de simplificación de los sistemas lógicos combinacionales 49
3.3 Funciones incompletas: Definición y aplicación de los métodos de simplificación .. 54
3.4 Multifunciones: Definición y aplicación de los métodos de simplificación 60
3.5 Realización de las funciones lógicas 65
3.5.1 Realización con puertas NO-Y (NAND) y NO-O (NOR) 65
3.5.2 Realización de funciones mediante el montaje «Y por conexión» 77
3.5.3 Realización de las funciones lógicas con puertas O-exclusiva 79

3.6 Fenómenos aleatorios en los sistemas combinacionales 79
3.6.1 Fenómenos aleatorios estáticos 80
3.6.2 Fenómenos aleatorios dinámicos 86
3.7 Bloques funcionales combinacionales. 89
3.7.1 Decodificadores. Demultiplexores 90
3.7.2 Codificadores 108
3.7.3 Multiplexores 112
3.7.3.1 Multiplexores realizados con puertas Y y puertas O 112
3.7.3.2 Multiplexores realizados con puertas de tres estados 121
3.7.3.3 Multiplexores realizados con interruptores 124
3.7.3.4 Aplicaciones de los multiplexores 130
3.7.4 Comparadores binarios 134
3.7.5 Detectores/generadores de paridad 136
3.8 Sistemas combinacionales programables 139
3.8.1 Sistemas combinacionales programables no universales 141
3.8.1.1 Matrices programables de puertas Y o decodificadores programables [Programmable gate arrays (PGA)] 141
3.8.1.2 Detectores de identidad (Identity comparators) 141
3.8.2 Sistemas combinacionales universales programables 144
3.8.2.1 Sistemas combinacionales universales programables completos. Memorias de acceso aleatorio 144
3.8.2.2 Sistemas combinacionales universales programables incompletos (SCUPI) 154
3.8.2.2.1 Matrices lógicas programables [Programmable logic array (PLA)] 155
3.8.2.2.2 Matrices lógicas Y-programables (PAL) 161
3.8.2.2.3 Ampliación de la capacidad de las matrices lógicas programables (PLA) y las matrices lógicas Y-programables (PAL) 169
3.8.2.2.4 Matrices lógicas de puertas universales 174

CAPITULO 4. Aritmética en los códigos binarios 179

4.1 Generalidades 179
4.2 Operaciones en el sistema binario natural 179
4.2.1 Suma aritmética binaria 179
4.2.2 El circuito sumador total como bloque funcional 191
4.2.3 Resta binaria 198
4.2.3.1 Representación de los números negativos 198
4.2.3.2 Operación con números negativos mediante el convenio del complemento a dos 199
4.2.3.3 Operación con números negativos mediante el convenio del complemento a uno 204
4.2.4 Unidades aritméticas y lógicas 208
4.2.5 Multiplicación binaria 211
4.3 Operaciones aritméticas en los códigos decimales codificados en binario (BCD) ... 216
4.3.1 Operaciones en el código decimal codificado en binario natural (BCD natural) 216
4.3.1.1 Suma en el código decimal codificado en binario natural (BCD natural) 216
4.3.1.2 Resta en el código decimal codificado en binario natural (BCD natural) 219
4.4 Formatos de representación de los números fraccionarios 225
4.4.1 Representación de los números fraccionarios en coma fija 227
4.4.2 Representación de los números fraccionarios en coma flotante 227

CAPITULO 5. Tecnologías de realización de los circuitos digitales 233

5.1 Introducción 233
5.2 Características generales de los circuitos digitales 233

5.3 Circuitos digitales con diodos ... 236
5.4 Circuitos digitales con transistores ... 238
5.4.1 Características principales del transistor bipolar en conmutación ... 238
5.4.2 Circuitos digitales con componentes discretos: lógica resistencia-transistor ... 242
5.4.3 Circuitos bloque: tecnología resistencia-transistor (RTL) ... 243
5.4.4 Circuitos integrados digitales ... 244
5.4.4.1 Introducción ... 244
5.4.4.2 Clasificación de los circuitos integrados digitales ... 244
5.4.4.2.1 Circuitos integrados digitales monolíticos ... 245
5.4.4.2.1.1 Clasificación según el tipo de dispositivo utilizado ... 245
5.4.4.2.1.2 Clasificación según el número de dispositivos ... 247
5.4.4.2.1.3 Clasificación de los circuitos normalizados o estándar ... 249
5.4.4.2.1.4 Circuitos especificados por el usuario o a medida ... 261
5.4.4.2.2 Circuitos integrados híbridos ... 266
5.4.4.3 Circuitos integrados monolíticos digitales bipolares de silicio ... 267
5.4.4.3.1 Circuitos integrados de lógica resistencia-transistor (RTL) ... 267
5.4.4.3.2 Circuitos integrados de lógica diodo-transistor (DTL) ... 267
5.4.4.3.3 Circuitos integrados de lógica transistor-transistor (TTL) ... 272
5.4.4.3.4 Circuitos integrados de lógica transistor-transistor (TTL) de tres estados ... 276
5.4.4.3.5 Tecnología de alta inmunidad al ruido ... 279
5.4.4.3.6 Circuitos digitales de lógica no saturada ... 281
5.4.4.3.6.1 Circuitos integrados de tecnología TTL Schottky 281
5.4.4.3.6.2 Circuitos integrados de lógica de acoplamiento por emisor (ECL) ... 283
5.4.4.3.7 Tecnología de inyección integrada (Integrated Injection Logic [IIL o I^2L]) ... 285
5.4.4.4 Circuitos integrados monolíticos digitales con transistores MOS de silicio ... 290
5.4.4.4.1 Introducción ... 290
5.4.4.4.2 Tecnologías digitales MOS ... 299
5.4.4.4.2.1 Circuitos integrados MOS estáticos ... 299
5.4.4.4.2.2 Circuitos integrados MOS dinámicos ... 301
5.4.4.4.2.3 Circuitos integrados CMOS ... 303
5.4.4.5 Circuitos integrados digitales con transistores bipolares y MOS (BIC-MOS) ... 309
5.4.4.6 Circuitos integrados digitales de arseniuro de galio (GaAs) ... 310
5.4.4.6.1 Introducción ... 310
5.4.4.6.2 Dispositivos de arseniuro de galio ... 311

CAPITULO 6. Sistemas secuenciales ... 315

6.1 Definición ... 315
6.2 Sistemas secuenciales asíncronos ... 320
6.2.1 Introducción ... 320
6.2.2 Sistemas secuenciales asíncronos caracterizados mediante niveles ... 321
6.2.2.1 Sistemas secuenciales asíncronos de realimentación directa ... 321
6.2.2.2 Biestables asíncronos activados mediante niveles ... 331
6.2.2.3 Síntesis de los sistemas secuenciales asíncronos con realimentación mediante biestables ... 333
6.2.3 Sistemas secuenciales asíncronos caracterizados mediante cambios de nivel ... 341
6.2.3.1 Especificaciones de transición de los sistemas secuenciales asíncronos caracterizados mediante cambios de nivel ... 341

6.2.3.2 Diagrama de secuencia de operaciones y grafo de transición de un autómata asíncrono de control 346
6.2.3.3 Biestables asíncronos accionados por los cambios de nivel 354
6.2.3.4 Síntesis de los sistemas secuenciales asíncronos con células asíncronas activadas por flancos (CAF) 357
6.2.3.5 Fenómenos aleatorios en los sistemas secuenciales asíncronos realizados con células activadas por flancos (CAF) 368
6.2.3.6 Problemática de la realización física de los sistemas secuenciales asíncronos 370
6.2.4 Circuitos digitales temporales 371
6.2.4.1 Introducción 371
6.2.4.2 Circuitos de memoria temporal 371
6.2.4.2.1 Monoestables 371
6.2.4.2.1.1 Monoestables no realimentados 373
6.2.4.2.1.2 Monoestables realimentados 374
6.2.4.2.2 Temporizadores analógico-digitales 383
6.2.4.3 Generadores de impulsos 385
6.2.4.3.1 Circuitos astables 385
6.2.4.3.2 Generadores de impulsos con temporizadores analógico-digitales 394
6.2.4.4 Aplicaciones de los circuitos digitales temporales 398
6.3 Sistemas secuenciales síncronos 402
6.3.1 Introducción 402
6.3.2 Biestables sincronizados 404
6.3.2.1 Biestables sincronizados por niveles 406
6.3.2.2 Biestables sincronizados por cambios de nivel 410
6.3.2.2.1 Biestables activados por impulsos 411
6.3.2.2.1.1 Biestables principal-subordinado (master-slave) 411
6.3.2.2.1.2 Biestables principal-subordinado con enclavamiento (master-slave with data lockout) 418
6.3.2.2.2 Biestables activados por flancos (edge-triggered) 420
6.3.3 Aplicación de los biestables sincronizados activados por flancos (edge triggered) a la síntesis de biestables asíncronos activados por flancos 423
6.3.4 Representación gráfica de los biestables síncronos 427
6.3.5 Sistemas secuenciales síncronos de control 430
6.3.5.1 Introducción 430
6.3.5.2 Sistemas secuenciales síncronos de control de aplicación general 437
6.3.5.2.1 Registros de entrada y salida en paralelo 437
6.3.5.2.2 Contadores síncronos 441
6.3.5.2.3 Registros de desplazamiento 447
6.3.5.3 Sistemas secuenciales síncronos asincronizados 448
6.3.5.4 Bloques funcionales síncronos 455
6.3.5.4.1 Introducción 455
6.3.5.4.2 Teoría general y símbolos normalizados 457
6.3.5.4.3 Contadores 470
6.3.5.4.3.1 Generalidades 470
6.3.5.4.3.2 Contadores asíncronos 471
6.3.5.4.3.3 Contadores síncronos 481
6.3.5.4.3.4 Aplicaciones de los contadores 490
6.3.5.4.3.4.1 Sistemas contadores y su visualización 490
6.3.5.4.3.4.2 Divisores de frecuencia 496
6.3.5.4.4 Registros de desplazamiento 496
6.3.5.4.4.1 Generalidades 496
6.3.5.4.4.2 Aplicaciones de los registros de desplazamiento 506
6.3.5.4.5 Registros de entrada y salida en paralelo 509
6.3.5.4.6 Conjuntos de registros 510
6.3.5.5 Síntesis de los sistemas secuenciales síncronos de control 516
6.3.5.5.1 Clasificación de los sistemas secuenciales síncronos 516

6.3.5.5.2 Síntesis de los sistemas secuenciales síncronos cableados 518
6.3.5.5.2.1 Síntesis de los sistemas secuenciales síncronos cableados con registros de entrada y salida en paralelo 518
6.3.5.5.2.2 Síntesis de los sistemas secuenciales síncronos cableados con contadores síncronos 522
6.3.5.5.3 Síntesis de los sistemas secuenciales síncronos microprogramables 530
6.3.5.5.3.1 Sistemas secuenciales síncronos microprogramables con circuitos combinacionales programables completos 531
6.3.5.5.3.2 Sistemas secuenciales síncronos microprogramables con circuitos combinacionales programables incompletos 541
6.3.5.5.4 Sistemas secuenciales síncronos modulares 552
6.3.5.5.5 Sistemas secuenciales síncronos semimodulares 556
6.3.5.6 Aplicaciones de los sistemas secuenciales síncronos de control 559
6.3.5.6.1 Introducción 559
6.3.5.6.2 Procesadores digitales secuenciales 559
6.3.5.6.3 Controladores lógicos síncronos 566
6.4 Representación gráfica de las señales de un sistema secuencial síncrono 570

CAPITULO 7. Unidades de memoria 577

7.1 Introducción 577
7.2 Parámetros y características más importantes de una memoria 578
7.2.1 Capacidad 578
7.2.2 Forma de acceder a la información 578
7.2.3 Estructura interna 579
7.2.3.1 Memorias de acceso aleatorio (Random Access Memories) 581
7.2.3.1.1 Características generales 581
7.2.3.1.2 Organización de las memorias de acceso aleatorio 585
7.2.3.1.3 Permanencia de la información 591
7.2.3.1.3.1 Memorias de escritura/lectura (R/W) o activas 591
7.2.3.1.3.1a Memorias activas de escritura y lectura no simultáneas 593
7.2.3.1.3.1b Memorias activas de escritura y lectura simultáneas 600
7.2.3.1.3.1c Memorias activas de acceso múltiple 604
7.2.3.1.3.2 Memorias pasivas 606
Memorias totalmente pasivas 608
Memorias pasivas programables 609
Memorias pasivas reprogramables 609
7.2.3.1.4 Diseño de unidades de memoria de acceso aleatorio con bloques funcionales 632
7.2.3.2 Memorias de acceso serie 636
7.2.3.2.1 Registros de desplazamiento 640
7.2.3.2.2 Memorias TUBO (FIFO) 649
7.2.3.2.3 Memorias PILA (LIFO) 655
7.2.3.3 Memorias asociativas (Content addressable memories) (CAM) 659
7.3 Tecnologías de las unidades de memoria 664
7.3.1 Introducción 664
7.3.2 Tecnologías de las memorias de acceso aleatorio (RAM) 664
7.3.2.1 Memorias activas 664
7.3.2.2 Memorias pasivas 676
7.3.3 Tecnología de las memorias de acceso serie 683

APENDICE 1. Símbolos lógicos normalizados 687

A1.1 Introducción 687
A1.2 Símbolo gráfico normalizado 688
A1.3 Símbolos asociados a las entradas y a las salidas 691
A1.4 Representación normalizada de los sistemas combinacionales 695
A1.4.1 Generalidades 695
A1.4.2 Puertas lógicas 696
A1.4.3 Símbolos de bloques funcionales combinacionales complejos 697
A1.4.3.1 Relación de dependencia Y [G (AND)] 698
A1.4.3.2 Relación de dependencia O [V (OR)] 700
A1.4.3.3 Relación de inversión [N (Negate)] 700
A1.4.3.4 Relación de desinhibición [EN (Enable)/Inhibición] 701
A1.4.3.5 Relación de modo de operación [M (Mode)] 702
A1.4.3.6 Relación de conexión (Z) 703
A1.4.3.7 Relación de direccionamiento [A (Address)] 703
A1.4.3.8 Combinación de relaciones de dependencia 704
A1.5 Representación normalizada de los sistemas secuenciales 706
A1.5.1 Generalidades 706
A1.5.2 Indicativos de los sistemas secuenciales 707
A1.5.3 Relaciones de dependencia en sistemas secuenciales 708
A1.5.3.1 Relaciones de puesta a cero (R) y puesta a uno (S) 708
A1.5.3.2 Relación de control (C) 709
A1.5.3.3 Relación de modo de operación (M) 712
A1.5.3.4 Combinación de relaciones de dependencia 712
A1.5.4 Ejemplos prácticos de sistemas secuenciales 717
A1.5.4.1 Registros de entrada y salida en paralelo 717
A1.5.4.2 Contadores 721
A1.5.4.2.1 Contadores asíncronos 721
A1.5.4.2.2 Contadores síncronos 722
A1.5.4.3 Registros de desplazamiento 727

APENDICE 2. Circuitos temporales analógico-digitales 731

A2.1 Introducción 731
A2.2 Circuito temporal multifuncional 555 731
A2.3 Circuito temporal multifuncional 556 736
A2.4 Circuito temporal analógico-digital 558 736

INDICE ALFABETICO 741

Prólogo a la octava edición

Es una satisfacción realizar el prólogo de la octava edición de un libro del que se han editado más de cuarenta mil ejemplares en los últimos veinticinco años.

Desde que en 1973 se publicó la primera edición de esta obra, la Electrónica Digital ha conocido un desarrollo extraordinario como consecuencia de la evolución de la Microelectrónica que pasó de la pequeña escala de integración (máximo de 10 puertas en un circuito integrado monolítico) en la década de 1960, hasta la giga-gran escala de integración (más de 100.000 puertas en un circuito integrado monolítico) a finales de la década de 1990. Esto ha obligado a una continua puesta al día, que se puede observar en esta nueva edición en la que se expone un método original de analizar los circuitos integrados digitales monolíticos y se estudian las nuevas formas de realizar los sistemas secuenciales síncronos.

La complejidad que están alcanzando algunas tecnologías, de lo cual la Electrónica Digital es un ejemplo característico, hace muy difícil la exposición en secuencia de los diferentes conceptos relativos a la misma. Por ello es de destacar en esta nueva edición la utilización de técnicas hipermedia para introducir al lector en los nuevos procedimientos de autoformación mediante la utilización de recursos informáticos que son una consecuencia a su vez del progreso de la Microelectrónica.

Esta obra, que fue en su momento pionera en España en la enseñanza de la Electrónica Digital mediante procedimientos tradicionales, es también pionera en la aplicación de las tecnologías de la información a la formación de técnicos. Por ello ha de resultar de gran interés para todos los que, además de estudiar Electrónica Digital, estén interesados en conocer las inmensas posibilidades que ofrece la enseñanza asistida por computador.

JUAN PERACAULA ROURA
Catedrático de Tecnología Electrónica
de la Universidad Politécnica de Cataluña

Introducción a la octava edición

Esta nueva edición es el resultado del esfuerzo del autor por incorporar al libro los cambios producidos en la Electrónica Digital en la década de 1990 e incorporar las nuevas técnicas de enseñanza asistida por computador a su análisis.

El vertiginoso desarrollo de los circuitos integrados monolíticos que en tan sólo treinta años han pasado de contener cien dispositivos electrónicos a más de un millón, ha hecho que se elevase enormemente la complejidad de los circuitos y sistemas electrónicos digitales y ha convertido a la Electrónica Digital en una tecnología compleja en la que resulta difícil formar a los especialistas mediante una exposición exclusivamente secuencial de los conceptos. Por otra parte, dicho desarrollo ha hecho que la práctica vaya por delante de la teoría y ha generado una falta de normalización de las denominaciones que cambian de unos fabricantes a otros.

Pero además en la década de 1990 la Electrónica Digital ha permitido el aumento de la capacidad de proceso y almacenamiento de los computadores, lo cual ha abierto la puerta a nuevos métodos de presentar la información tanto de gráficos como de textos a través de lo que se conoce como las técnicas hipermedia.

La evolución de la Electrónica Digital se puede seguir a través de las ocho ediciones del libro «Sistemas Electrónicos Digitales» que han ido incorporando de forma paulatina los nuevos conceptos, tratando de crear una teoría que armonizase el desarrollo y facilitase la formación de expertos.

Esta nueva edición sigue la senda de las siete anteriores pero además incorpora las técnicas hipermedia para facilitar al estudiante la autoevaluación de sus conocimientos y la formación interactiva en los nuevos conceptos complejos surgidos a partir de los circuitos integrados monolíticos de ultra-gran y giga-gran escalas de integración (ULSI y GLSI). Dicha incorporación ha sido llevada a cabo en equipo entre el grupo de Informática Educativa de la Universidad de las Villas de Cuba y el

Instituto de Electrónica Aplicada Pedro Barrié de la Maza de la Universidad de Vigo, y ha sido propiciada por un proyecto de enseñanza asistida por computador de la Electrónica subvencionado por el Instituto de Cooperación Iberoamericana. Por ello el autor debe expresar su agradecimiento al codirector del citado proyecto, profesor Víctor Giraldo Valdés Pardo, cuya experiencia y dedicación han sido decisivas para alcanzar los resultados que se presentan en el CD-ROM incluido en el libro y a la profesora María Alicia Valdés Peña que programó y diseño gráficamente el libro hipermedia de autoformación. Igualmente el autor debe agradecer a la profesora del departamento de Tecnología Electrónica de la Universidad de Vigo María Dolores Valdés Peña el diseño y la dirección de la programación de la aplicación hipermedia de clasificación y descripción de los circuitos integrados digitales.

Para realizar esta nueva edición ha sido necesario eliminar el estudio de los circuitos que han dejado de utilizarse en la práctica y describir los nuevos circuitos resultado del desarrollo de la Microelectrónica. En esa tarea ha sido de gran valía el trabajo en equipo con diversos miembros del departamento de Tecnología Electrónica de la Universidad de Vigo entre los que debo citar muy especialmente a María José Moure Rodríguez y Vicente Rodríguez Vázquez.

Como aspectos más importantes de esta octava edición cabe citar:

- Su división en dos tomos de los que el primero incluye los diferentes circuitos electrónicos digitales tanto combinacionales como secuenciales y, en particular, las memorias digitales y el segundo los procesadores digitales secuenciales y muy especialmente los programables.
- La creación de una nueva clasificación de los sistemas secuenciales síncronos de acuerdo con su arquitectura, que los divide en no modulares, semimodulares y modulares.
- La creación de una nueva forma de clasificar los circuitos integrados digitales monolíticos que incluye los nuevos tipos de circuitos y muy especialmente los circuitos digitales configurables.
- La inclusión de un CD-ROM que contiene dos sistemas hipermedia:
  - — Un hiperdocumento que contiene la clasificación de los circuitos integrados digitales monolíticos anteriormente indicada y permite al lector acceder a la información de forma interactiva no secuencial. La gran cantidad de imágenes y gráficos asociados al documento hacen que pueda ser utilizado para el aprendizaje autodidacta.
  - — Un sistema de autoevaluación de la asimilación de los principales conceptos relacionados con los circuitos combinacionales, los circuitos secuenciales y las memorias.

Por todo ello esta nueva edición constituye un sistema de gran utilidad para la formación de los ingenieros eléctricos y electrónicos en las características y las aplicaciones de los sistemas electrónicos digitales.

La acogida dispensada a ediciones anteriores ha sido el mejor estímulo para tratar de hacer el esfuerzo necesario para presentar esta nueva edición.

EL AUTOR

Capítulo

1

Sistemas y códigos de numeración

1.1 GENERALIDADES

Los sistemas digitales actúan bajo el control de variables discretas, entendiéndose por éstas, las variables que pueden tomar un número finito de valores. Por ser de fácil realización los componentes físicos con dos estados diferenciados, es éste el número de valores utilizado usualmente para dichas variables que, por lo tanto, son binarias.

Tanto si se utilizan en proceso de datos (Informática), transmisión de información (Comunicaciones de datos o Telemática) o en sistemas de control (Electrónica Industrial), los sistemas electrónicos digitales realizan operaciones con variables discretas que constituyen números o caracteres alfabéticos. Los números pueden representarse en diversos sistemas de numeración, que se diferencian por su base. La base de un sistema de numeración es el número de símbolos distintos utilizados para la representación de las cantidades en el mismo. El sistema de numeración utilizado en la vida cotidiana es el de base diez, en el cual existen diez símbolos distintos, del 0 al 9.

Por la razón expuesta el sistema de numeración más utilizado en la realización de los sistemas digitales es el de base dos, o binario, en el cual existen solamente dos símbolos, que son el 0 y el 1.

Estudiaremos en este capítulo los sistemas de numeración, dedicando especial atención al sistema binario y las diferentes formas de codificar la información en dicho sistema.

1.2 REPRESENTACION DE LOS NUMEROS. SISTEMAS DE NUMERACION

En un sistema de base b, un número N cualquiera se puede representar mediante un polinomio de potencias de la base, multiplicadas por un símbolo perteneciente al sistema.

En general tendremos:

$$N = a_n b^n + a_{n-1} b^{n-1} + \ldots + a_i b^i + \ldots + a_0 b^0 + a_{-1} b^{-1} + \ldots + a_{-p} b^{-p}$$

siendo b la base del sistema de numeración y a_i un número perteneciente al sistema y que, por tanto, cumple la condición $0 \leq a_i < b$; $n + 1$ y p representan respectivamente el número de digitos enteros y fraccionarios.

Si el sistema es de base diez o decimal, tendremos $b = 10$ y $0 \leq a_i < 10$. Por ejemplo, el número 87,54 en base diez se representa por:

$$87{,}54 = 8 \cdot 10^1 + 7 \cdot 10^0 + 5 \cdot 10^{-1} + 4 \cdot 10^{-2}$$

En el sistema de base ocho u octal se tiene $b = 8$ y $0 \leq a_i < 8$ y el número 673,54 se representa por:

$$673{,}54 = 6 \cdot 8^2 + 7 \cdot 8^1 + 3 \cdot 8^0 + 5 \cdot 8^{-1} + 4 \cdot 8^{-2}$$

De igual forma, en el sistema de base dos o binario, se tiene $b = 2$ y $0 \leq a_i < 2$ y el número 1011,11 en este sistema se representa por el polinomio:

$$1011{,}11 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

Para identificar el sistema al cual pertenece un número se suele indicar la base como subindice. De esta forma los números mencionados como ejemplo se representarán:

$$87{,}54_{10};\ 673{,}54_8;\ 1011{,}11_2$$

1.3 SISTEMA BINARIO

Como ya hemos expuesto, este sistema utiliza solamente dos símbolos distintos, que se representan gráficamente por 0 y 1 y reciben el nombre de bit. La utilización casi exclusiva de este sistema de numeración en los equipos de cálculo y control automáticos es debida a la seguridad y rapidez de respuesta de los elementos físicos que poseen dos estados diferenciados y a la sencillez de las operaciones aritméticas en este sistema, las cuales compensan la necesidad de utilizar mayor número de cifras (para representar una misma cantidad) que en los sistemas cuya base es mayor de dos.

La conversión de un número entero de base dos a decimal se realiza fácilmente representando el número mediante su polinomio equivalente y operando éste en base diez. Un ejemplo aclarará la regla:

$$\begin{aligned} 1101{,}11_2 &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = \\ &= 8 + 4 + 1 + 1/2 + 1/4 = 13 + 0{,}5 + 0{,}25 = 13{,}75_{10} \end{aligned}$$

Por tanto:

$$1101{,}11_2 = 13{,}75_{10}$$

En la tabla 1.1 se representa la equivalencia entre los dieciséis primeros números enteros del sistema binario y decimal.

Se demuestra fácilmente que, si se divide un número entero expresado en un sistema de base $b_1 > b_2$ por la base b_2, y el cociente se vuelve a dividir por b_2 y así sucesivamente, el último cociente y los restos obtenidos forman el número en el sistema de base b_2.

Sistema binario	*Sistema decimal*
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	10
1 0 1 1	11
1 1 0 0	12
1 1 0 1	13
1 1 1 0	14
1 1 1 1	15

TABLA 1.1

En efecto, sea el número N en la base b_1 que suponemos ya convertido a la base b_2 y expresado mediante su polinomio equivalente:

$$(N)_{b_1} = a_n b_2^n + a_{n-1} b_2^{n-1} + \ldots + a_1 b_2^1 + a_0 b_2^0$$

Dividiendo ambos miembros de esta ecuación por la base b_2 resulta:

$$\frac{(N)_{b_1}}{b_2} = a_n\, b_2^{n-1} + a_{n-1}\, b_2^{n-2} + \ldots + a_1 + \frac{a_0}{b_2}$$

De esto se deduce que, al dividir $(N)_{b_1}$ por b_2, el resto obtenido a_0 es igual al dígito menos significativo del número representado en base b_2.

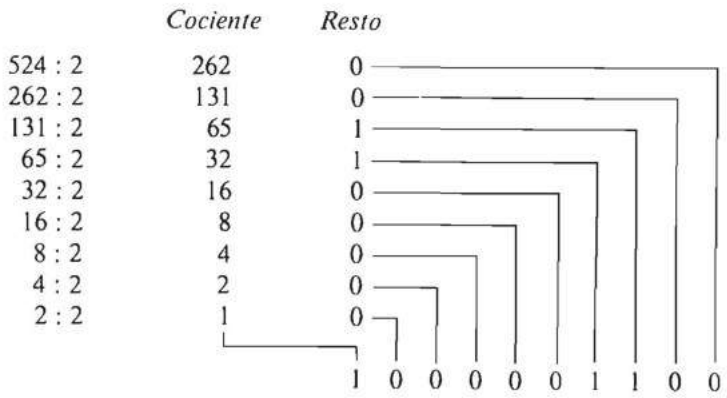
Designando al cociente $a_n b_2^{n-1} + a_{n-1} b_2^{n-2} + \ldots + a_1$ por N_1 y dividiendo nuevamente por b_2, resulta:

$$\frac{N_1}{b_2} = a_n\, b_2^{n-2} + a_{n-1}\, b_2^{n-3} + \ldots + \frac{a_1}{b_2}$$

donde el resto a_1 representa el segundo dígito de N en la base b_2.

Continuando el proceso se llegan a obtener todos los dígitos de N en la base b_2, siendo el más significativo el último cociente.

Como ejemplo realizaremos la conversión del número 524 en base diez a la base dos.



Luego:

$$524_{10} = 1000001100_2$$

Para convertir un número fraccionario en la base b_1 a la base b_2 partiremos de la igualdad:

$$(N)_{b_1} = a_{-1}b_2^{-1} + a_{-2}b_2^{-2} + \ldots + a_{-n}b_2^{-n}$$

Multiplicando ambos miembros por b_2 resulta:

$$b_2(N)_{b_1} = a_{-1} + a_{-2}b_2^{-1} + \ldots + a_{-n}b_2^{-n+1}$$

La parte entera obtenida a_{-1} representa la cifra más significativa del número N en base b_2. Si la parte fraccionaria resultante se vuelve a multiplicar por b_2, el nuevo cociente es la segunda cifra de N en base b_2.
En efecto, escribiendo:

$$M = a_{-2}b_2^{-1} + \ldots + a_{-n}b_2^{-n+1}$$

y multiplicando por b_2 ambos miembros resulta:

$$b_2 \cdot M = a_{-2} + \ldots + a_{-n}b_2^{-n+2}$$

Continuando el proceso se obtienen todas las cifras de N en base b_2. Este proceso terminará cuando la parte fraccionaria obtenida sea nula.

Aplicando el método expuesto convertiremos el número 0,825 en base diez a la base dos.

$$\begin{aligned}
0{,}825 &\cdot 2 = 1{,}650 \\
0{,}65 &\cdot 2 = 1{,}30 \\
0{,}30 &\cdot 2 = 0{,}6 \\
0{,}6 &\cdot 2 = 1{,}2 \\
0{,}2 &\cdot 2 = 0{,}4 \\
0{,}4 &\cdot 2 = 0{,}8 \\
0{,}8 &\cdot 2 = 1{,}6
\end{aligned}$$

de donde resulta:

$$0{,}825_{10} = 0{,}1101001\ldots_2$$

1.4 SISTEMA OCTAL

El interés de este sistema de numeración, en el cual existen ocho símbolos diferentes (del 0 al 7), proviene de que la conversión de los números al sistema de base dos y viceversa resulta muy sencilla por ser $8 = 2^3$.

Para convertir un número en base ocho a la base dos se convierte cada cifra en su equivalente binario. Un ejemplo aclarará el método. Sea el número 325,6 en base ocho. Se tiene:

$$3_8 = 011_2$$
$$2_8 = 010_2$$
$$5_8 = 101_2$$
$$6_8 = 110_2$$

Por tanto:

$$325{,}6_8 = 011010101{,}110_2 = 11010101{,}11_2$$

La conversión del sistema de base dos al sistema de base ocho se realiza a la inversa agrupando los bits enteros y fraccionarios en grupos de tres a partir de la coma decimal. Para completar el último grupo se añadirán los ceros necesarios. Sea, por ejemplo, el número 11010,1011 en base dos. Añadiendo un cero a la izquierda y dos a la derecha obtenemos los siguientes grupos de tres bits cuyo equivalente octal indicamos al lado:

	Octal
011	3
010	2
101	5
100	4

Resulta, pues,

$$11010{,}1011_2 = 32{,}54_8$$

La conversión de un número en el sistema de base ocho al de base 10 y viceversa se realiza mediante los métodos generales indicados en el apartado 1.3. Como ejemplo obtendremos el equivalente decimal del número 354 en base ocho.

$$354_8 = 3 \cdot 8^2 + 5 \cdot 8^1 + 4 \cdot 8^0 = 192 + 40 + 4 = 236_{10}$$

Para obtener el equivalente octal del número decimal 1036:

	Cociente	*Resto*
1036 : 8	129	4
129 : 8	16	1
16 : 8	2	0
		2 0 1 4

Resulta:

$$1036_{10} = 2014_8$$

1.5 SISTEMA HEXADECIMAL

El sistema de numeración hexadecimal es el de base dieciséis, es decir, para la representación de las cantidades utiliza dieciséis símbolos diferentes que son los dígitos del 0 al 9 y las letras del alfabeto de la A a la F.

El interés de este sistema, al igual que el del octal, es debido a que 16 es una potencia de 2($2^4 = 16$), y por lo tanto resulta muy sencilla la conversión de los números del sistema binario natural al hexadecimal y viceversa.

En la tabla 1.2 se representa la combinación binaria equivalente a cada uno de los símbolos del sistema hexadecimal.

Sistema binario natural	*Sistema hexadecimal*
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	A
1 0 1 1	B
1 1 0 0	C
1 1 0 1	D
1 1 1 0	E
1 1 1 1	F

TABLA 1.2

Para convertir un número del sistema hexadecimal al binario se sustituye cada símbolo por su equivalente en binario indicado en la tabla 1.2. Sea por ejemplo el número $9A7E_{16}$. El equivalente de cada símbolo es:

$$9_{16} = 1\ 0\ 0\ 1_2$$
$$A_{16} = 1\ 0\ 1\ 0_2$$
$$7_{16} = 0\ 1\ 1\ 1_2$$
$$E_{16} = 1\ 1\ 1\ 0_2$$

Por lo tanto resulta:

$$9A7E_{16} = 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0_2$$

La conversión de un número del sistema binario natural al hexadecimal se realiza a la inversa agrupando los bits enteros y fraccionarios en grupos de cuatro a partir de la coma decimal y convirtiendo cada grupo independientemente. Para completar el último grupo se añaden los ceros que sean necesarios. Sea por ejemplo el número 100111, 10101 en base dos. Añadiendo dos ceros a la izquierda y tres a la derecha resulta:

$$\begin{aligned} 0\ 0\ 1\ 0_2 &= 2_{16} \\ 0\ 1\ 1\ 1_2 &= 7_{16} \\ 1\ 0\ 1\ 0_2 &= A_{16} \\ 1\ 0\ 0\ 0_2 &= 8_{16} \end{aligned}$$

Resulta por lo tanto:

$$1\ 0\ 0\ 1\ 1\ 1, 1\ 0\ 1\ 0\ 1_2 = 27, A\ 8_{16}$$

1.6 CODIGOS BINARIOS

Se entiende por código una representación unívoca de las cantidades y/o de los caracteres alfabéticos de tal forma que, a cada uno de ellos, se asigna una combinación de símbolos determinada y viceversa.

De esta definición se deduce que los sistemas de numeración estudiados en los apartados anteriores constituyen códigos de representación de cantidades. El sistema binario recibe el nombre de código binario natural. Este código presenta grandes ventajas en la realización de las operaciones aritméticas tal como se estudia en el capítulo 4 y por ello es el que se utiliza, casi exclusivamente, en la realización de procesadores digitales programables que se estudian en el tomo 2. Sin embargo, existen muchos sistemas de control digitales en los que, a causa de sus características peculiares, se utilizan otros códigos binarios distintos del binario natural.

Con n cifras binarias o bits se pueden obtener 2^n combinaciones diferentes y cada una de ellas se puede asignar a una cantidad distinta. Por ello el número posible de asignaciones es el de permutaciones de las 2^n combinaciones, es decir $2^n!$, que constituyen otros tantos códigos binarios. De entre todos ellos se estudian en apartados sucesivos aquellos cuya utilización es más frecuente debido a que poseen alguna propiedad particular.

1.6.1 Códigos decimales codificados en binario

La información procesada por cualquier sistema digital ha de convertirse finalmente al sistema decimal para que pueda ser interpretada con mayor facilidad. Esta es la principal razón de la existencia de los códigos decimales codificados en binario [en inglés «Binary Coded Decimal» (BCD)] que, como se analiza en el capítulo 3, se convierten muy fácilmente al sistema decimal.

En los códigos BCD cada número decimal se codifica directamente en un código binario. Para representar los diez dígitos del cero al nueve se necesitan 4 bits porque con 3 bits sólo se pueden codificar 2^3 dígitos, es decir, del 0 al 7. Por tanto, de las 16 (2^4) combinaciones posibles con 4 bits, en los códigos BCD solamente se utilizan diez. Esto hace que el número de bits necesarios para representar una cantidad en un código BCD sea superior al del código binario natural y por ello cuando en la década de los sesenta todavía no existían memorias de acceso aleatorio realizadas con transistores, los códigos BCD se utilizaban para diseñar sistemas electrónicos de control y calculadoras digitales de bolsillo o de sobremesa de pequeña capacidad. En la actualidad son numerosos los procesadores digitales que poseen una unidad operativa con capacidad para realizar operaciones en el código binario natural y en el BCD natural que se describe seguidamente.

Los códigos BCD se pueden clasificar en dos grandes clases que son los BCD ponderados y los BCD no ponderados, que se estudian a continuación.

a) Códigos BCD ponderados.

Los códigos BCD ponderados son aquellos en los que a cada posición o cifra binaria se le asigna un peso y el número decimal equivalente a una combinación binaria se obtiene sumando los pesos de las posiciones que poseen el valor uno. Así pues, un número decimal N del 0 al 9 se representa por la expresión:

$$N = \sum_{i=0}^{i=3} p_i x_i$$

en la cual x_i puede tomar el valor 0 o el 1 y p_i es el peso de la posición i.

Existen diversos códigos BCD ponderados, tres de los cuales se indican en la tabla 1.3. El único de ellos que se utiliza en la actualidad es el BCD natural, en el cual los pesos de las distintas posiciones coinciden con los del código binario natural, es decir, son 1, 2, 4 y 8. Tal como se puede observar en la tabla 1.3, los números decimales del 0 al 9 se codifican en BCD natural con las mismas combinaciones de ceros y unos que en binario natural. Las combinaciones 1010 a 1111 no se utilizan.

El código BCD Aiken (tabla 1.3) presenta la propiedad de ser autocomplementario, es decir, es un código en el que la combinación correspondiente al complemento a 9 de N ($9-N$), se obtiene invirtiendo la combinación correspondiente a N, o sea, cambiando los ceros por unos y viceversa. En efecto, el número 3 se codifica mediante la combinación 0011 y el 6 ($9-3$) mediante 1100. Esta propiedad hace que un sumador/restador de números codificados en BCD Aiken se realice con un circuito electrónico más sencillo que un sumador/restador de números codificados en BCD natural. Este hecho tenía una gran importancia cuando no existían circuitos integrados monolíticos e incluso cuando éstos sólo habían alcanzado los niveles de pequeña (SSI) y mediana (MSI) escala de integración (ver capítulo 5). El progreso de la microelectrónica ha hecho que la menor complejidad del sumador/restador en BCD Aiken no suponga ninguna ventaja en la práctica y ha producido el abandono de la utilización de dicho código. Esto constituye un ejemplo de como el avance de la microelectrónica no solamente ha permitido hacer los sistemas digitales más pequeños sino que ha producido un cambio drástico en los procedimientos de diseño.

Dígito decimal	BCD natural P_3 P_2 P_1 P_0 8 4 2 1	BCD Aiken P_3 P_2 P_1 P_0 2 4 2 1	P_3 P_2 P_1 P_0 5 4 2 1
0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0	0 1 0 0
5	0 1 0 1	1 0 1 1	1 0 0 0
6	0 1 1 0	1 1 0 0	1 0 0 1
7	0 1 1 1	1 1 0 1	1 0 1 0
8	1 0 0 0	1 1 1 0	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 0 0

TABLA 1.3.—**Códigos BCD ponderados.**

b) Códigos BCD no ponderados.

Como su nombre indica son aquellos en los que cada posición binaria no tiene asignado un peso. El más conocido de todos ellos es el BCD exceso tres representado en la tabla 1.4. En este código cada número decimal N se codifica mediante la combinación que le corresponde a $N+3$ en el BCD natural. El código BCD exceso tres posee la propiedad de ser autocomplementario como el código BCD Aiken y, por la misma razón que este último, el progreso de la microelectrónica ha hecho que casi no se utilice.

Dígito decimal	BCD exceso tres
0	0 0 1 1
1	0 1 0 0
2	0 1 0 1
3	0 1 1 0
4	0 1 1 1
5	1 0 0 0
6	1 0 0 1
7	1 0 1 0
8	1 0 1 1
9	1 1 0 0

TABLA 1.4.—**Código BCD exceso tres.**

La conversión de un número decimal a un código BCD se realiza simplemente expresando cada dígito mediante la combinación que le corresponde en el código de que se trate. Como ejemplo a continuación se representa el número decimal 748 en los códigos BCD natural, exceso tres y Aiken:

BCD natural:	0 1 1 1	0 1 0 0	1 0 0 0
BCD exceso tres:	1 0 1 0	0 1 1 1	1 0 1 1
BCD Aiken:	1 1 0 1	0 1 0 0	1 1 1 0

1.6.2 Códigos binarios continuos y cíclicos

Un código binario es continuo si las combinaciones correspondientes a números decimales consecutivos son adyacentes. Se denominan combinaciones binarias adyacentes a aquellas que difieren solamente en un bit.

Un código continuo en el que la última combinación es adyacente a la primera se denomina cíclico.

La simple observación de la tabla 1.1 y la tabla 1.3 hace ver que tanto el código binario natural como el BCD natural no son continuos.

El código binario cíclico de mayor difusión es el reflejado, que recibe el nombre de código Gray y se representa en la tabla 1.5, junto al binario natural, para núme-

Dígito decimal	*Código binario natural*	*Código Gray*
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1
10	1 0 1 0	1 1 1 1
11	1 0 1 1	1 1 1 0
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0

TABLA 1.5.—**Código Gray y binario natural.**

2 bits	3 bits	4 bits
0 0	0 0 0	0 0 0 0
0 1	0 0 1	0 0 0 1
1 1	0 1 1	0 0 1 1
1 0	0 1 0	0 0 1 0
	1 1 0	0 1 1 0
	1 1 1	0 1 1 1
	1 0 1	0 1 0 1
	1 0 0	0 1 0 0
		1 1 0 0
		1 1 0 1
		1 1 1 1
		1 1 1 0
		1 0 1 0
		1 0 1 1
		1 0 0 1
		1 0 0 0

TABLA 1.6

ros de 4 bits. La denominación de reflejado es debida a que la formación de un código de n bits se realiza partiendo del de $n-1$ bits, repitiendo simétricamente las combinaciones de éste y añadiendo por la izquierda un nuevo bit 0 para las 2^{n-1} primeras combinaciones y 1 para las 2^{n-1} siguientes. En la tabla 1.6 se representa la formación de los códigos Gray de 2, 3 y 4 bits. Una gran ventaja de este código es su facilidad de conversión al código binario natural y viceversa mediante la función O-exclusiva que se estudia en el apartado 2.5.1.

Los códigos continuos se utilizan en todas aquellas aplicaciones en las que alguna imperfección puede hacer que aparezcan combinaciones erróneas debido a que no cambian simultáneamente todos los bits que deben hacerlo.

El ejemplo más característico de aplicación de los códigos continuos es la medida de desplazamientos longitudinales y angulares mediante un codificador óptico absoluto como el representado en la figura 1.1. Dicho codificador consiste en un disco acoplado al eje del sistema mecánico cuyo desplazamiento se desea medir. Dicho disco está dividido en un conjunto de pistas que, a su vez, se dividen en sectores. Cada pista está formada por una sucesión de zonas opacas y transparentes y tiene asignado un emisor de luz y un receptor de luz enfrentados entre sí y situados a uno y otro lado del disco respectivamente. El receptor de luz está construido de tal forma que, cuando recibe luz, su salida se encuentra a un nivel de tensión correspondiente al uno lógico y, cuando no la recibe, a un nivel correspondiente al cero lógico. Si el número de pistas es n, el disco se puede dividir en 2^n sectores y, si a cada sector se le asigna una combinación de zonas opacas y transparentes diferentes, el disco con sus emisores y receptores constituye un codificador óptico. La necesidad de utilizar un código cí-

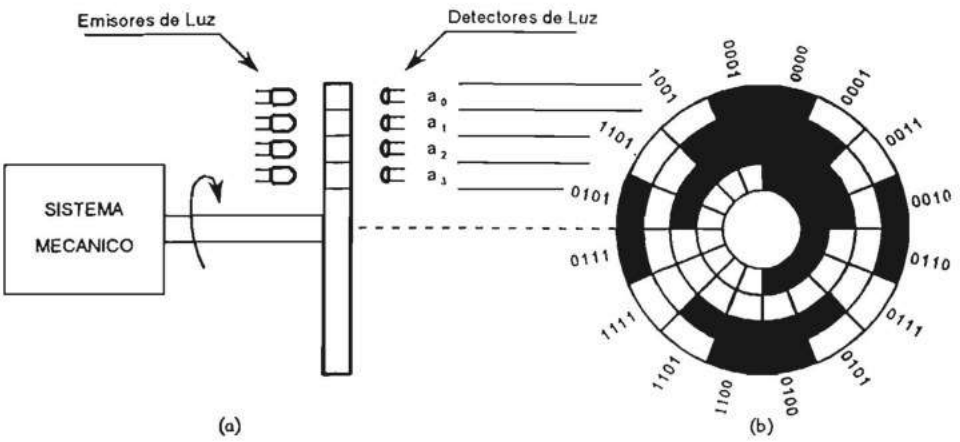


FIGURA 1.1.—**Aplicación del código Gray: *a*) Sistema mecánico acoplado a un codificador absoluto de posición; *b*) vista frontal del disco codificador.**

clico es debida a que alguna imperfección en la construcción de las zonas opacas y transparentes, una desalineación de los emisores y receptores o algún desajuste en el sistema mecánico cuyo eje se une al codificador, pueden hacer que aparezca una combinación intermedia errónea en el caso de que el código utilizado no lo sea. En la figura 1.1 se representa un codificador óptico con cuatro pistas que dan lugar a dieciséis sectores codificados en el código binario Gray.

Otro ejemplo de código continuo y cíclico es el progresivo Johnson representado en la tabla 1.7 para combinaciones de 5 bits. La capacidad de codificación de un código Johnson de n posiciones binarias es de $2n$ cantidades diferentes. Por ejemplo, mediante 5 bits, se pueden representar solamente los números del 0 al 9. Dado que la

Digito decimal	*Código Johnson*
0	0 0 0 0 0
1	0 0 0 0 1
2	0 0 0 1 1
3	0 0 1 1 1
4	0 1 1 1 1
5	1 1 1 1 1
6	1 1 1 1 0
7	1 1 1 0 0
8	1 1 0 0 0
9	1 0 0 0 0

TABLA 1.7.—**Código progresivo Johnson.**

máxima capacidad de codificación con n bits es 2^n, este código no se utiliza en sistemas digitales complejos por implicar una mayor complejidad de los mismos. Tal como se estudia en el capítulo 6, la gran sencillez de los contadores cuyos estados se codifican en el código Johnson hace posible su utilización en la unidad de control de numerosos procesadores digitales. Esto tuvo un gran interés cuando el estado incipiente de la microelectrónica en la década de los sesenta impedía hacer los bloques funcionales síncronos con entradas de inhibición y era conveniente garantizar la ausencia de impulsos aleatorios al decodificar los estados de un contador. En la actualidad se utiliza para realizar contadores cuyo contenido es transferido, en cualquier instante, a un procesador secuencial síncrono.

1.7 CODIGOS ALFANUMERICOS

En los códigos estudiados en los apartados anteriores solamente era posible representar información numérica. Pero en muchos sistemas digitales, tanto de control como de proceso de datos, es necesario representar información alfabética y además algunos signos especiales, lo que ha dado lugar a la existencia de códigos alfanuméricos.

De entre los diversos códigos alfanuméricos existentes, ha sido definido como código internacional el ASCII (del inglés «American Standard Code for Information

Bits 4 3 2 1 \ 6 5	0 0	0 1	1 0	1 1
0 0 0 0	@	P	b	0
0 0 0 1	A	Q	!	1
0 0 1 0	B	R	»	2
0 0 1 1	C	S	#	3
0 1 0 0	D	T	$	4
0 1 0 1	E	U	%	5
0 1 1 0	F	V	&	6
0 1 1 1	G	W	'	7
1 0 0 0	H	X	(	8
1 0 0 1	I	Y	)	9
1 0 1 0	J	Z	*	:
1 0 1 1	K	[	+	;
1 1 0 0	L	\	,	<
1 1 0 1	M	]	−	=
1 1 1 0	N	↑	.	>
1 1 1 1	O	←	/	?

b = equivale a espacio en blanco

TABLA 1.8.—**Código ASCII de 6 bits.**

Interchange»). Para representar los 26 caracteres alfabéticos y los 10 numéricos hacen falta 6 bits porque $2^5 < 36 < 2^6$. De esta forma se dispone de 28 combinaciones, hasta las 64 posibles, para codificar los símbolos especiales.

En la tabla 1.8 se representa el código ASCII de 6 bits. Los números del 0 al 9 se codifican en BCD natural mediante los bits 1 al 4.

Además, en sistemas de proceso de datos es interesante codificar directamente, mediante una combinación binaria, algunas órdenes correspondientes a instrucciones de operación de unidades periféricas tales como impresoras, sistemas de transmisión de datos, etc. Para ello se añade un séptimo bit, obteniéndose de esta forma el código ASCII completo.

1.8 CODIGOS DETECTORES DE ERRORES

En el manejo, y especialmente en la transmisión de una información numérica, es posible que se produzcan errores debido a la presencia de ruido en el proceso o por avería de alguno de los componentes.

Cuando en un código binario se utilizan todas las combinaciones posibles de sus n posiciones, es imposible la detección de un error, porque una combinación del código se transforma en otra que también pertenece a él. Por consiguiente, la detección de errores en un código binario se logra no utilizando todas las combinaciones posibles. Pero esta condición, aunque es necesaria, no es suficiente para que el código permita detectar errores. Por ejemplo, el código BCD natural no utiliza más que diez combinaciones de las dieciséis posibles de los cuatro bits, pero si por un error de un bit, la combinación 0011 se convierte en 0111, no es posible detectarlo, porque ambas combinaciones pertenecen al código.

Para establecer la condición necesaria y suficiente para que un código binario permita detectar errores es necesario definir el concepto de distancia mínima de un código. La distancia entre dos combinaciones binarias viene dada por el número de bits de una de ellas que deben ser modificados para obtener la otra. En un código se define la distancia mínima como la menor de las distancias entre dos combinaciones binarias cualesquiera pertenecientes al mismo. El valor de la distancia mínima de los códigos estudiados hasta ahora (ver tablas 1.1, 1.3, 1.4, 1.5 y 1.7) es la unidad y, por tanto, un error en uno solo de los bits de un número binario perteneciente a cualquiera de ellos, puede convertirlo en otro número perteneciente al mismo y hacer que el error no sea detectable.

De todo lo dicho se deduce que, para que un código pueda detectar errores, su distancia mínima ha de ser superior a la unidad.

Existen diversos tipos de códigos detectores de errores, entre los cuales se encuentran los códigos de peso constante (entendiendo por peso de una combinación binaria el número de unos lógicos de la misma) y los códigos de paridad constante. Ambos tipos de códigos se analizan a continuación.

a) Códigos de peso constante.

Entre los códigos de peso constante se encuentran el 2 entre 5 y el biquinario. En la tabla 1.9 se presenta el código 2 entre 5 en el cual cada combinación binaria posee

Dígito decimal	*Código 2 entre 5*
0	0 1 1 0 0
1	1 1 0 0 0
2	1 0 1 0 0
3	1 0 0 1 0
4	0 1 0 1 0
5	0 0 1 1 0
6	1 0 0 0 1
7	0 1 0 0 1
8	0 0 1 0 1
9	0 0 0 1 1

TABLA 1.9.—**Código 2 entre 5.**

solamente dos unos lógicos. Se observa también que la distancia mínima de este código es 2.

En la tabla 1.10 se representa el código biquinario, que es un código ponderado de 7 bits cuya distancia mínima también es dos y que ha sido utilizado directamente en operaciones aritméticas.

La detección de errores en los códigos de peso constante se puede realizar mediante un circuito detector de paridad estudiado en el apartado 3.7.5 del capítulo 3.

Las ventajas que presentan los códigos de paridad constante, que se estudian a continuación, hacen que los códigos detectores de errores de peso constante no se utilicen en la actualidad.

Dígito decimal	*Peso* 5 0 4 3 2 1 0
0	0 1 0 0 0 0 1
1	0 1 0 0 0 1 0
2	0 1 0 0 1 0 0
3	0 1 0 1 0 0 0
4	0 1 1 0 0 0 0
5	1 0 0 0 0 0 1
6	1 0 0 0 0 1 0
7	1 0 0 0 1 0 0
8	1 0 0 1 0 0 0
9	1 0 1 0 0 0 0

TABLA 1.10.—**Código biquinario.**

b) Códigos de paridad constante.

Los códigos de paridad constante se obtienen añadiendo a las combinaciones de los códigos de distancia unidad anteriormente estudiados un bit llamado de paridad. Si el código que se desea obtener es de paridad par, dicho bit debe ser tal que el número de unos en cada combinación del nuevo código sea par. Si, por el contrario, el código resultante es de paridad impar, el bit añadido a cada combinación ha de ser tal que la resultante tenga un número impar de unos.

La detección de errores en estos códigos consiste en comprobar si el número de unos de cada combinación es par (códigos de paridad par) o impar (códigos de paridad impar). En el apartado 3.7.5 del capítulo 3 se estudian los circuitos lógicos que permiten la generación del bit de paridad y la detección de la paridad de una combinación binaria.

En la tabla 1.11 se representa el código decimal codificado en binario natural con un bit de paridad impar. Se observa que la distancia mínima de las combinaciones pertenecientes a este código es dos.

Tanto los códigos de peso constante como los de paridad constante son de distancia mínima dos y permiten, por lo tanto, la detección de errores de un bit. Para poder detectar errores de más de un bit es necesaria la utilización de un código de distancia mínima superior a dos. En general, el número de bits erróneos que se pueden detectar es igual al número en que la distancia mínima es superior a la unidad.

La posibilidad de utilizar códigos detectores de errores de paridad constante, obtenidos a partir de los códigos de distancia mínima igual a la unidad como el binario natural o el BCD natural, constituye una de las características de los sistemas electrónicos digitales que los hacen idóneos para la transmisión de información a distancia a través de un canal serie en el que las señales eléctricas se puedan alterar debido a la presencia de ruido.

Digito decimal	*BCD natural*	*Bit de paridad impar*
0	0 0 0 0	1
1	0 0 0 1	0
2	0 0 1 0	1
3	0 0 1 1	1
4	0 1 0 0	0
5	0 1 0 1	1
6	0 1 1 0	1
7	0 1 1 1	0
8	1 0 0 0	0
9	1 0 0 1	1

TABLA 1.11.—**Código decimal codificado en binario natural con bit de paridad.**

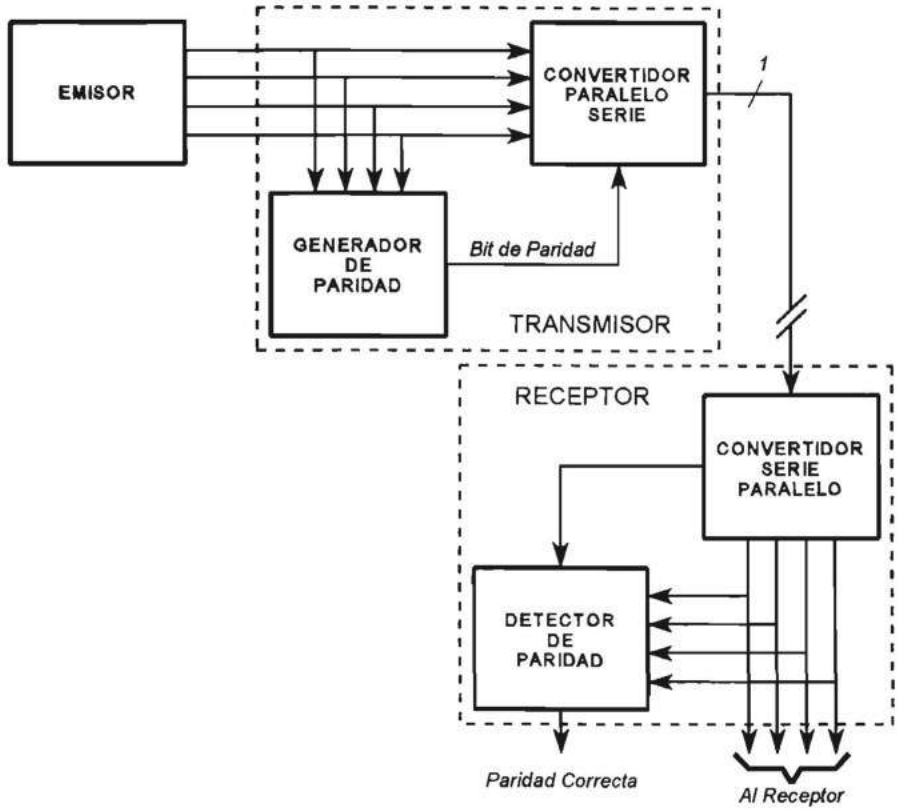


FIGURA 1.2.—**Diagrama de bloques básico de un sistema de comunicación digital.**

En la figura 1.2 se representa el diagrama de bloques básico de un sistema de comunicación digital. El emisor genera información en un código de paridad no constante (como, por ejemplo, el binario natural o el BCD natural). Dicha información se aplica a la entrada de un circuito que genera el bit de paridad y este último, junto con la primera, se aplica a un convertidor paralelo-serie que lo transmite a través de un único canal (hilo, fibra óptica, etc.) en cuyo extremo se coloca un sistema receptor formado por un convertidor serie-paralelo que, como su nombre indica, vuelve a colocar la información en paralelo. La salida de este convertidor se aplica a un circuito detector de paridad cuya salida indica si la paridad de la combinación recibida es o no correcta.

1.9 CODIGOS CORRECTORES DE ERRORES

Los códigos correctores de errores no sólo indican la existencia de un error sino que proporcionan información de cuál es la cifra o cifras binarias erróneas y, por consiguiente, permiten su corrección simplemente invirtiendo el bit correspondiente.

Estos códigos se utilizan solamente en la transmisión de información, especialmente cuando no es posible volver a enviarla en caso de producirse un error. Esto sucede, en general, en algunos sistemas digitales que trabajan en tiempo real como, por ejemplo, aquellos que están situados en un satélite y actúan sobre ciertos elementos del mismo a partir de los valores de determinadas variables del proceso.

Los códigos de distancia mínima dos estudiados en el apartado anterior no permiten la corrección de errores, porque al producirse un error la combinación obtenida posee como mínimo dos adyacentes pertenecientes al código y no es posible discernir de cuál de las dos procede. Por ejemplo, si en el código de la tabla 1.9 de distancia mínima dos se detecta la combinación errónea 10001, es imposible conocer si el error se ha producido en el primer bit y la combinación correcta es 10000 o en el segundo y la combinación correcta es 10011.

Por lo tanto, para poder corregir errores, la distancia mínima del código ha de ser superior a dos. Si la distancia mínima de un código es tres, la combinación obtenida por error en un bit es adyacente a una sola combinación del código y es posible conocer cuál es el bit erróneo. Así, un código de distancia mínima tres permite detectar errores de dos bits o corregir errores de un bit. En general, la distancia mínima de un código para que permita corregir errores de n bits ha de ser $d_m = 2n + 1$.

Nos limitaremos a estudiar los códigos correctores de errores de un bit cuya distancia mínima es tres y entre ellos los de mayor difusión, que son los códigos de Hamming.

Los códigos de Hamming están basados en la adición a un código de distancia unidad de n bits, como los estudiados en el apartado 1.6, de p bits, obteniéndose un nuevo código de $n + p$ bits. En este nuevo código se realizan p detecciones de paridad en bits seleccionados del mismo, obteniéndose un bit de paridad uno o cero según el número de bits sea par o impar. El conjunto de los p bits de paridad forma un número en el sistema binario natural cuyo equivalente decimal nos indica la posición del bit erróneo. En caso de que no exista error, dicho número debe de ser cero. El número p de bits añadidos ha de ser suficiente para permitir la detección de error y la de ausencia de error en las $n + p$ posiciones. Dado que con p bits se obtienen 2^p combinaciones, se ha de cumplir la relación $2^p \geqslant n + p + 1$.

Como ejemplo realizaremos el código de Hamming obtenido a partir del código decimal codificado en binario natural. En este código $n = 4$ y, por tanto, el número de bits que se han de añadir es 3, dado que $2^3 = 4 + 3 + 1$. Para detectar los siete posibles errores de un bit en cada una de las posiciones y la ausencia de error son necesarias ocho combinaciones binarias que denominaremos correctoras de error. Dichas combinaciones se obtienen mediante 3 bits c_1, c_2 y c_3 y el número decimal equivalente al binario formado por ellos ha de indicar la posición errónea.

Veremos ahora la forma de generar cada uno de los bits de la combinación correctora de errores. En la tabla 1.12 se presentan todas las combinaciones de los bits c_1, c_2 y c_3.

El bit c_1 ha de tomar el valor 1 si se produce un error en los bits b_1, b_3, b_5 y b_7 de la combinación del código. Si el número de unos existentes en esas cuatro posiciones es siempre par, un error en uno cualquiera de esos cuatro bits lo convierte en impar. Por tanto c_1 ha de valer uno si el número de unos en las posiciones

N.° decimal equivalente	c_3	c_2	c_1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

TABLA 1.12

b_1, b_3, b_5 y b_7 es impar, y cero en caso contrario. En el apartado 2.5.1 del capítulo 2 se estudia la función lógica O-exclusiva cuyo valor lógico es uno en las condiciones especificadas. Luego algebraicamente:

$$c_1 = b_1 \oplus b_3 \oplus b_5 \oplus b_7$$

donde $\oplus$ es el símbolo de la función O-exclusiva.

De igual forma se deduce que c_2 y c_3 han de obtenerse por medio de las expresiones:

$$c_2 = b_2 \oplus b_3 \oplus b_6 \oplus b_7$$
$$c_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7$$

para lo cual ha de cumplirse la condición de que el número de unos ha de ser par en las combinaciones b_2, b_3, b_6, b_7 y b_4, b_5, b_6, b_7.

Para lograr estas condiciones se han de generar adecuadamente los tres bits que se añaden a los cuatro de la combinación BCD.

Dado que b_1, b_2 y b_4 sólo aparecen en una expresión cada uno, los elegiremos como bits añadidos a la combinación BCD formada por b_3, b_5, b_6 y b_7. El bit b_1 ha de valer uno si el número de unos de b_3, b_5 y b_7 es impar y cero en caso contrario: por tanto:

$$b_1 = b_3 \oplus b_5 \oplus b_7$$

De igual forma b_2 y b_4 se han de obtener respectivamente:

$$b_2 = b_3 \oplus b_6 \oplus b_7$$
$$b_4 = b_5 \oplus b_6 \oplus b_7$$

De todo lo anterior se deduce el código de Hamming presentado en la tabla 1.13.

Como ejemplo comprobaremos la detección de un error en el bit b_5 de la combinación 0011001 correspondiente al número decimal 2; la combinación errónea es 0001001.

N.º decimal equivalente	b_7	b_6	b_5	b_4	b_3	b_2	b_1
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	0	1	1	0	0	1
3	0	0	1	1	1	1	0
4	0	1	0	1	0	1	0
5	0	1	0	1	1	0	1
6	0	1	1	0	0	1	1
7	0	1	1	0	1	0	0
8	1	0	0	1	0	1	1
9	1	0	0	1	1	0	0

TABLA 1.13

Para detectarlo comprobaremos el valor lógico de c_1, c_2 y c_3

$$c_1 = b_1 \oplus b_3 \oplus b_5 \oplus b_7 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$
$$c_2 = b_2 \oplus b_3 \oplus b_6 \oplus b_7 = 0 \oplus 0 \oplus 0 \oplus 0 = 0$$
$$c_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

En efecto, la combinación c_3, c_2, c_1 es 101, equivalente al número decimal 5.

PROBLEMAS

1. Convertir al sistema binario el número decimal 1576.
2. Convertir al sistema octal el número binario 0111011.
3. Convertir al sistema decimal el número octal 576.
4. Convertir al sistema decimal el número binario 1101101,001.
5. Convertir al sistema octal el número decimal 25.740.
6. Convertir al sistema binario el número decimal 875,62.
7. Convertir al sistema decimal el número octal 1024,75.
8. Obtener el número equivalente en BCD exceso tres del decimal 5274.
9. Obtener el número decimal equivalente al número 0110 1000 0100 en BCD natural.
10. Indicar cuál es la distancia mínima del código BCD Aiken. Obtener a partir de él un código de paridad par.
11. Obtener un código de Hamming corrector de errores de un bit a partir del código BCD exceso tres.
12. Convertir el número 1100 1000 0011 perteneciente al código BCD exceso tres a:
    a) el código BCD natural;
    b) el código BCD Aiken;
    c) el sistema binario natural;
    d) el sistema decimal.

Capítulo

2

Algebra de Boole

2.1 DEFINICION Y POSTULADOS

Un álgebra de Boole es toda clase o conjunto de elementos que pueden tomar dos valores perfectamente diferenciados, que designaremos por 0 y 1 y que están relacionados por dos operaciones binarias denominadas suma (+) y producto (.)* lógicos que cumplen los siguientes postulados:

a) Ambas operaciones son conmutativas, es decir si a y b son elementos del álgebra, se verifica:

$$a + b = b + a; \quad a \cdot b = b \cdot a$$

b) Dentro del álgebra existen dos elementos neutros, el 0 y el 1, que cumplen la propiedad de identidad con respecto a cada una de dichas operaciones:

$$0 + a = a; \quad 1 \cdot a = a$$

c) Cada operación es distributiva con respecto a la otra:

$$a \cdot (b + c) = a \cdot b + a \cdot c; \quad a + b \cdot c = (a + b) \cdot (a + c)$$

d) Para cada elemento, a, del álgebra existe un elemento denominado, $\bar{a}$, tal que:

$$a + \bar{a} = 1; \quad a \cdot \bar{a} = 0$$

Este postulado define realmente una nueva operación fundamental que es la inversión o complementación de una variable. La variable $\bar{a}$ se encuentra siempre en un estado binario contrario al de a.

La primera ecuación expresa la imposibilidad de que a y $\bar{a}$ tomen el valor lógico cero al mismo tiempo y la segunda ecuación indica que nunca pueden tener el valor lógico uno al mismo tiempo. Por lo tanto la tabla de verdad de la inversión o complementación es:

a	$\bar{a}$
0	1
1	0

**Nota:* La operación producto se indica en general simplemente mediante la ausencia de símbolo entre dos variables.

Puede elegirse como postulado un grupo distinto del adoptado con tal de que se cumpla la condición de que ninguno pueda ser deducido de cualquiera de los demás.

De lo explicado anteriormente se deduce que el álgebra de Boole es un ente matemático. En realidad, físicamente son varios los conjuntos que poseen dos operaciones binarias que cumplen los postulados desarrollados. Ejemplo de estos conjuntos son el álgebra de las proposiciones o juicios formales y el álgebra de la conmutación formada también por elementos que pueden tomar dos estados perfectamente diferenciados. Estos elementos son los circuitos lógicos cuyo estudio desarrollaremos en capítulos sucesivos.

Los primeros circuitos de conmutación o lógicos utilizados han sido los contactos y, aunque poco a poco han sido desplazados por los circuitos electrónicos, pueden ser empleados para memorizar más fácilmente las leyes del álgebra de Boole antes expresadas y los teoremas que se desarrollan seguidamente.

La operación suma se asimila a la conexión en paralelo de contactos y la operación producto a la conexión en serie. El inverso de un contacto es otro cuyo estado es siempre el opuesto del primero, es decir está cerrado cuando aquél está abierto y viceversa. El elemento 0 es un contacto que está siempre abierto y el elemento 1 un contacto que está siempre cerrado. Además se considera una función de transmisión entre los dos terminales de un circuito de contactos, que toma el valor 1, cuando existe un camino para la circulación de corriente entre ellos (cortocircuito) y el valor 0 al no existir dicho camino (circuito abierto).

En la figura 2.1, se expresa gráficamente que el álgebra de los contactos cumple las leyes del álgebra de Boole.

2.2 TEOREMAS DEL ALGEBRA DE BOOLE

Basándose en los postulados anteriores se deducen los teoremas que expondremos seguidamente. Su demostración se puede realizar algebraicamente o mediante la llamada tabla de verdad. La tabla de verdad de una expresión algebraica binaria representa los valores que dicha expresión puede tomar para cada combinación de estados de las variables que forman parte de la misma. Dos expresiones algebraicas que tienen la misma tabla de verdad son equivalentes.

Teorema 1: Cada identidad deducida de los anteriores postulados del álgebra de Boole permanece válida si la operación «+» y «·» y los elementos 0 y 1 se intercambian entre sí.

Este principio, llamado de dualidad, se deduce inmediatamente de la simetría de los cuatro postulados con respecto a ambas operaciones y a ambos elementos neutros.

Teorema 2: Para cada elemento a de un álgebra de Boole se verifica:

$$a + 1 = 1 \text{ y } a \cdot 0 = 0$$

Demostraremos la primera igualdad y con ello quedará demostrada por dualidad la segunda. En efecto, se verifica:

$$1 = a + \bar{a} = a + \bar{a} \cdot 1 = (a + \bar{a}) \cdot (a + 1) = 1 \cdot (a + 1) = a + 1$$

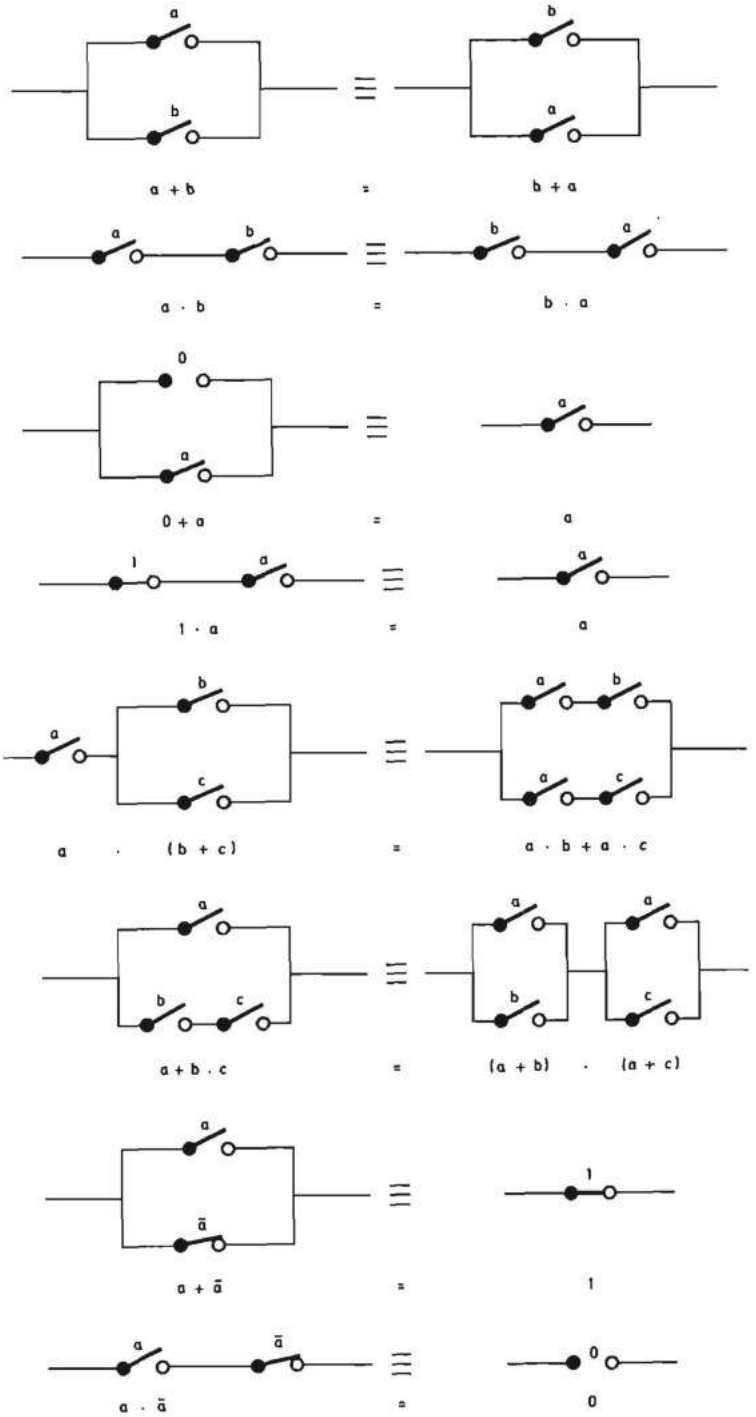


FIGURA 2.1.—**Representación de las leyes del álgebra de Boole mediante asociación de contactos.**

De este teorema y del postulado *b*) se deducen las siguientes igualdades:

$$0 + 0 = 0 \qquad 0 \cdot 0 = 0$$
$$0 + 1 = 1 \qquad 0 \cdot 1 = 0$$
$$1 + 1 = 1 \qquad 1 \cdot 1 = 1$$

y, por tanto, las tablas de verdad de las operaciones lógicas suma (función O) y producto (función Y) son las siguientes:

a	b	s
0	0	0
0	1	1
1	0	1
1	1	1

$s = a + b$

a	b	p
0	0	0
0	1	0
1	0	0
1	1	1

$p = a \cdot b$

Teorema 3: Para cada elemento a de un álgebra de Boole se verifica:

$$a + a = a \quad \text{y} \quad a \cdot a = a$$

Demostraremos la primera igualdad:

$$a = a + 0 = a + a\bar{a} = (a + a) \cdot (a + \bar{a}) = a + a$$

Teorema 4: Para cada par de elementos de un álgebra de Boole a y b, se verifica:

$$a + ab = a \quad \text{y} \quad a\,(a + b) = a$$

Esta ley se llama de absorción.

a	b	$a+ab$
0	0	0
0	1	0
1	0	1
1	1	1

TABLA 2.1

Lo demostraremos algebraicamente y mediante la tabla de verdad. En efecto, algebraicamente:

$$a = 1 \cdot a = (1 + b)\,a = 1 \cdot a + ab = a + ab$$

En la tabla 2.1 se comprueba que la columna correspondiente a $a + ab$ es igual a la columna de la variable a y por tanto se deduce la igualdad:

$$a = a + ab$$

Teorema 5: En un álgebra de Boole, las operaciones suma y producto son asociativas

$$a + (b + c) = (a + b) + c = a + b + c$$
$$a\,(bc) = (ab)\,c = abc$$

Este teorema se demuestra fácilmente mediante la tabla de verdad.

Teorema 6: Para todo elemento $\bar{a}$ de un álgebra de Boole se verifica:

$$\bar{\bar{a}} = a$$

Su demostración es inmediata mediante la tabla de verdad, lo cual se verifica seguidamente

a	$\bar{a}$	$\bar{\bar{a}}$
0	1	0
1	0	1

De este teorema y el postulado *d)* se deduce que en toda álgebra de Boole:

$$\bar{0} = 1 \quad \text{y} \quad \bar{1} = 0$$

Teorema 7: En toda álgebra de Boole se verifica:

(1) $\overline{a + b + c + d + \ldots} = \bar{a}\,\bar{b}\,\bar{c}\,\bar{d}\ldots$

(2) $\overline{a\,b\,c\,d\ldots} = \bar{a} + \bar{b} + \bar{c} + \bar{d}\ldots$

Demostraremos la primera de estas igualdades, denominadas leyes de De Morgan, con lo cual la segunda quedará demostrada por dualidad.

Realizaremos primero la demostración para dos variables.

De acuerdo con el postulado *d)*, para que $\overline{a + b} = \bar{a}\bar{b}$ se ha de verificar:

$$(a + b)\,\bar{a}\bar{b} = 0$$
$$(a + b) + \bar{a}\bar{b} = 1$$

En efecto, aplicando los postulados y teoremas ya estudiados resulta

$$(a + b)\,\bar{a}\bar{b} = a\bar{a}\bar{b} + b\bar{a}\bar{b} = 0 + 0 = 0$$
$$(a + b) + \bar{a}\bar{b} = (a + b + \bar{a})\,(a + b + \bar{b}) = 1 \cdot 1 = 1$$

La generalización para un número cualquiera de variables resulta ahora muy sencilla:

Denominaremos $b + c + d + \ldots = p$ y aplicando la ley de De Morgan que hemos demostrado para dos variables, resulta:

$$\overline{a + b + c + d} + \ldots = \overline{a + p} = \bar{a}\bar{p} = \bar{a}\overline{b + c + d + \ldots}$$

Llamando ahora $q = c + d + \ldots$ resulta:

$$\overline{a + b + c + d + \ldots} = \bar{a}\overline{b + q} = \bar{a}\bar{b}\bar{q} = \bar{a}\bar{b}\overline{c + d + \ldots}$$

Repitiendo este proceso se obtiene:

$$\overline{a + b + c + d + \ldots} = \bar{a}\bar{b}\bar{c}\bar{d}\ldots$$

c.q.d.

Este teorema define realmente dos nuevas funciones lógicas de gran importancia que, como veremos en el capítulo dedicado a estudiar la tecnología, serán utilizadas como elementos básicos para la realización de los sistemas digitales. Estas dos funciones, que realizan las expresiones (1) y (2), se denominan respectivamente NO-O (NOR) y NO-Y (NAND).

Las tres funciones elementales, suma, producto, e inversión lógica, pueden ser realizadas mediante las funciones NO-Y y NO-O.

En efecto, aplicando el teorema de De Morgan tenemos:

$$ab = \overline{\overline{ab}} = \overline{\bar{a} + \bar{b}}$$

$$a + b = \overline{\overline{a + b}} = \overline{\bar{a}\bar{b}}$$

y la inversión se realiza con una función NO-O o NO-Y de una sola entrada.

Para representar las funciones lógicas fue necesario crear símbolos adecuados. El desarrollo de la electrónica digital ha sido tan rápido que hizo que se crearan símbolos sin un estudio minucioso. Inicialmente se adoptaron símbolos diferentes para cada función, de los que son un ejemplo los de las funciones Y y O representados en la figura 2.2.

FIGURA 2.2.—**Símbolos lógicos no normalizados de las funciones Y y O.**

La búsqueda de una simplificación en la representación de los elementos lógicos llevó a la realización de estudios por parte de asociaciones de ingeniería (IEEE) y organizaciones de normalización (ISO) que han cristalizado en la adopción de un sistema normalizado de representación por parte de la Comisión Electrotécnica Internacional. Al estudio de este sistema se dedica el apéndice 1 de este libro. El nuevo sistema internacional llegó después de más de diez años durante los cuales los fabricantes de semiconductores crearon sus propios símbolos. Por ello se ha considerado lo más conveniente en esta edición combinar adecuadamente la simbología antigua y la nueva para que el lector pueda adaptarse de forma paulatina al cambio que supone pasar de una a otra.

FIGURA 2.3.—**Símbolos lógicos normalizados de las funciones Y y O.**

En la figura 2.3 se representan los símbolos de las puertas Y y O en el nuevo sistema.

En los símbolos antiguos la inversión unida a otras funciones se puede representar mediante un círculo; por tanto, los símbolos de la función NO-O (NOR) y NO-Y (NAND) se deducen respectivamente de los de las funciones O e Y añadiéndoles un círculo (fig. 2.4)

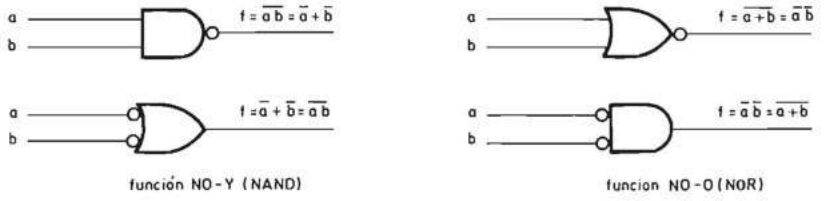


FIGURA 2.4.—**Símbolos lógicos no normalizados de las funciones NO-Y (NAND) Y NO-O (NOR).**

En el nuevo sistema la inversión unida a otras funciones se puede representar mediante un círculo pero en el caso de utilizar el criterio de lógica positiva (descrito en el apartado 5.1) se sustituye por un triángulo rectángulo (ver apéndice 1). En este libro se utiliza el círculo y por ello se representan las puertas NO-Y (NAND) y NO-O (NOR) mediante los símbolos de la figura 2.5.

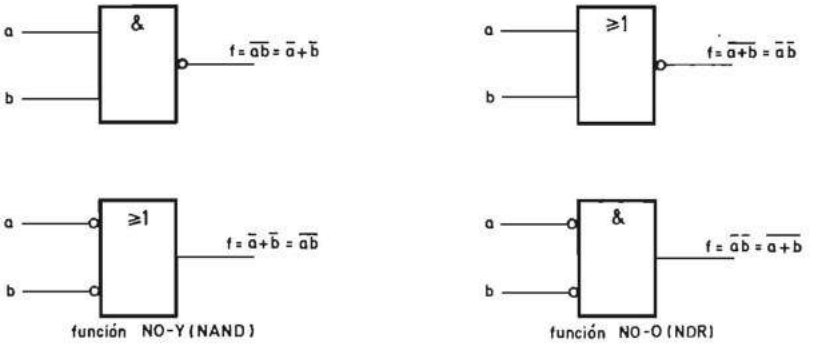


FIGURA 2.5.—**Símbolos lógicos normalizados de las funciones NO-Y (NAND) y NO-O (NOR).**

El teorema de De Morgan indica que existen dos formas de expresar la función NO-O (NOR) y la función NO-Y (NAND):

$$\overline{a + b} = \bar{a}\,\bar{b}$$

$$\overline{ab} = \bar{a} + \bar{b}$$

La segunda expresión de la función NO-O se puede representar mediante el símbolo de la función Y precedido de dos inversiones. Igualmente la función NO-Y se puede representar mediante el símbolo de la función O precedido de dos inversiones. En las figuras 2.4 y 2.5 se indican ambos símbolos.

Las funciones NO-O (NOR) y NO-Y (NAND) de una sola variable constituyen la función de inversión, por lo que esta función se puede representar mediante el símbolo de cualquiera de ellas con una sola variable de una entrada o mediante un símbolo especial. En la simbología antigua la función inversión se representa mediante un triángulo seguido de un círculo, o una puerta NO-Y o NO-O de una entrada tal como se indica en la figura 2.6. En el nuevo sistema normalizado se representa mediante el símbolo de un seguidor (un 1 como indicativo) con el triángulo o círculo de inversión a la salida (figura 2.7)

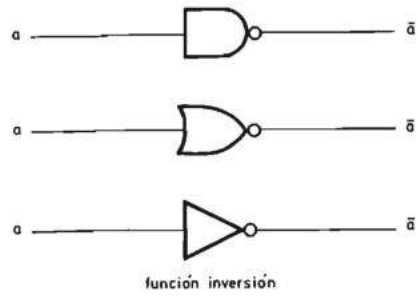


FIGURA 2.6.—**Símbolos lógicos no normalizados de un inversor.**

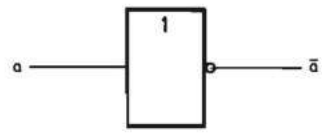


FIGURA 2.7.—**Símbolo lógico normalizado de un inversor.**

La realización de las funciones suma, producto e inversión con las funciones NO-Y y NO-O se representan gráficamente, mediante los símbolos antiguos en la figura 2.8 y los nuevos normalizados en la figura 2.9.

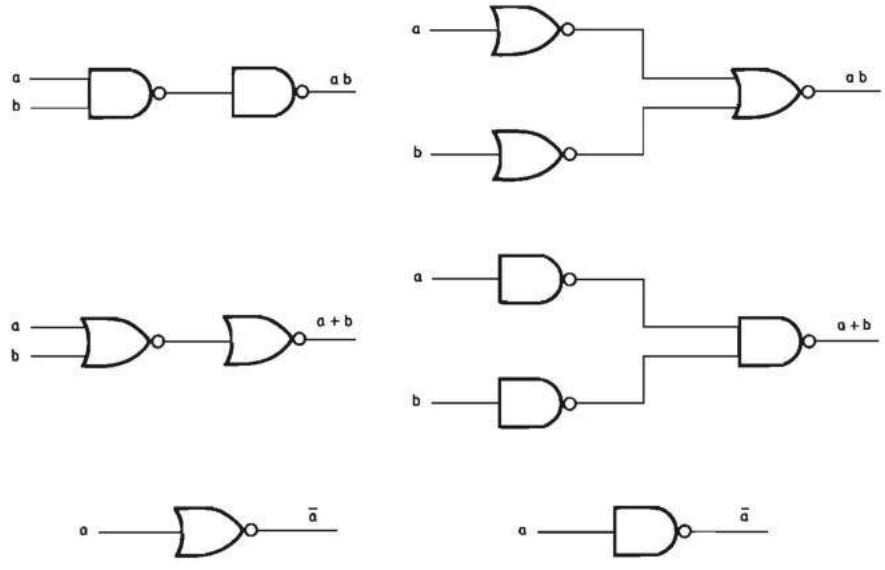


FIGURA 2.8.—**Esquemas de la realización de las funciones producto, suma e inversión con funciones NO-Y (NAND) y NO-O (NOR), representadas con símbolos no normalizados.**

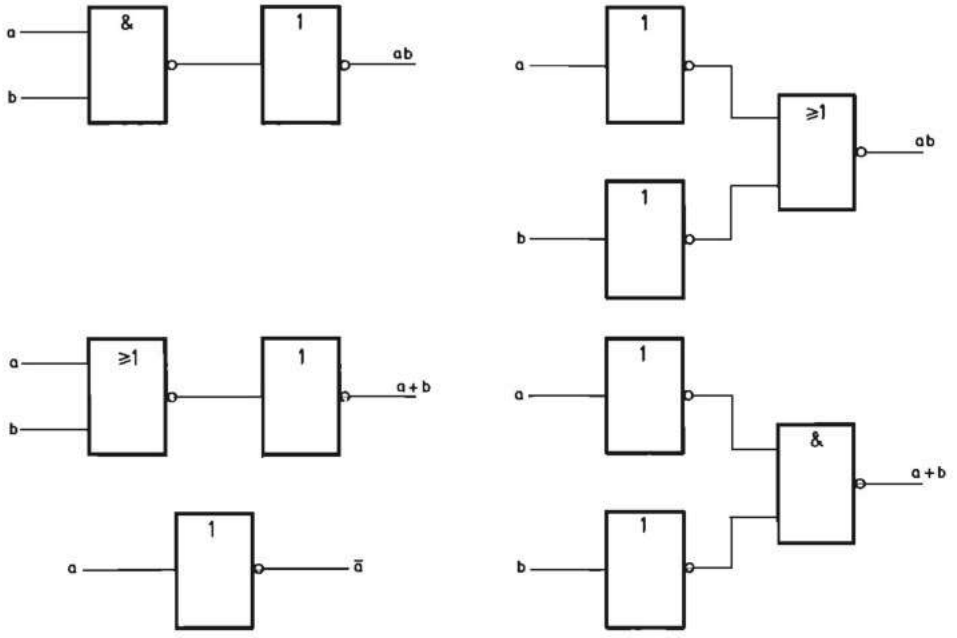


FIGURA 2.9.—**Esquemas de la realización de las funciones producto, suma e inversión con funciones NO-Y (NAND) y NO-O(NOR) representadas mediante símbolos normalizados.**

2.3 FUNCION DE UN ALGEBRA DE BOOLE

Una función de un álgebra de Boole es una variable binaria cuyo valor es igual al de una expresión algebraica en la que se relacionan entre sí las variables binarias por medio de las operaciones básicas, Producto lógico, Suma lógica e Inversión.

Se representa una función lógica por la expresión $f = f(a, b, c, ...)$

El valor lógico de f, depende del de las variables a, b, c, ...

Se llama término canónico de una función lógica a todo producto o suma en la cual aparecen todas las variables en su forma directa o inversa. Al primero de ellos se le llama producto canónico (minterm) y al segundo suma canónica (maxterm).

Por ejemplo sea una función de tres variables $f(a, b, c)$. El término $a\bar{b}c$ es un producto canónico y el término $\bar{a} + b + \bar{c}$ es una suma canónica.

El número máximo de productos canónicos o sumas canónicas viene dado por las variaciones con repetición de dos elementos tomados de n en n. El número de productos o sumas canónicas de n variables es por lo tanto 2^n.

Para mayor facilidad de representación, cada término canónico, se expresa mediante un número decimal equivalente al binario obtenido al sustituir las variables ordenadas con un criterio determinado por un 1 o un 0 según aparezcan en su forma directa o complementada respectivamente.

Por ejemplo, los términos canónicos siguientes se representarán:

$$\bar{d}cb\bar{a} \equiv 0110_2 = 6_{10}$$
$$d + \bar{c} + b + \bar{a} \equiv 1010_2 \equiv 10_{10}$$

Por lo tanto la función lógica $f(a, b, c) = \bar{a}b\bar{c} + ab\bar{c} + a\bar{b}c$ se podrá representar por la expresión: $f(a, b, c) = \Sigma_3(2, 3, 5)$ en la cual el símbolo Σ representa la suma lógica.

De igual forma la función $f(a, b, c) = (a + \bar{b} + \bar{c})(\bar{a} + b + \bar{c})(a + b + c)$ se puede representar por $f(a, b, c) = //_3(1, 2, 7)$ en la cual $//$ indica el producto lógico.

Cuando una función se expresa como una suma de productos canónicos o un producto de sumas canónicas se dice que se encuentra en su forma canónica.

Demostraremos ahora un teorema relativo a las funciones de un álgebra de Boole de gran importancia en la simplificación algebraica de las funciones lógicas.

Teorema. Toda función de un álgebra de Boole se puede expresar de la siguiente forma:

$$f(a, b, c...) = af(1, b, c, ...) + \bar{a}f(0, b, c...)$$
$$f(a, b, c...) = [a + f(0, b, c...)][\bar{a} + f(1, b, c...)]$$

Demostraremos la primera ecuación y la segunda quedará también demostrada por dualidad.

Para demostrarla es suficiente comprobar que la igualdad se cumple tanto para $a = 0$ como para $a = 1$.

En efecto si $a = 0$ y $\bar{a} = 1$; se verifica:

$$f(a, b, c...) = f(0, b, c...) = 0f(1, b, c...) + 1f(0, b, c...) = f(0, b, c...)$$

y si $a = 1$ y $\bar{a} = 0$; se verificará así mismo:

$$f(a, b, c...) = f(1, b, c...) = 1f(1, b, c...) + 0f(0, b, c...) = f(1, b, c...)$$

Quedan por lo tanto demostradas ambas igualdades. Multiplicando la primera de ellas por *a y por* $\bar{a}$ se obtienen respectivamente las relaciones:

$$af(a, b, c...) = af(1, b, c...)$$
$$\bar{a}f(a, b, c...) = \bar{a}f(0, b, c...)$$

Igualmente sumando a y $\bar{a}$ a la segunda igualdad se obtiene:

$$a + f(a, b, c...) = a + f(0, b, c...)$$
$$\bar{a} + f(a, b, c...) = \bar{a} + f(1, b, c...)$$

Estas últimas cuatro expresiones se pueden utilizar para simplificar algebraicamente las funciones lógicas. Por ejemplo dada la función:

$$f = abc + \bar{a}(b + a\bar{c} + \bar{a}\bar{b}\bar{c})$$

resulta aplicando la segunda igualdad al segundo sumando:

$$f = abc + \bar{a}(b + \bar{b}\bar{c})$$

El teorema que acabamos de demostrar permite llegar a la conclusión de que

toda función lógica puede transformarse en una función canónica bajo cualquiera de las dos formas anteriormente indicadas.

En efecto según hemos demostrado:

$f(a, b, c...) = a\,f(1, b, c...) + \bar{a}\,f(0, b, c...)$ y dado que:
$f(1, b, c...) = bf(1, 1, c...) + \bar{b}\,f(1, 0, c...)$ y
$f(0, b, c...) = bf(0, 1, c...) + \bar{b}\,f(0, 0, c...)$ resulta
$f(a, b, c...) = ab\,f(1, 1, c...) + a\bar{b}\,f(1, 0, c, ...) + \bar{a}\,b\,f(0, 1, c...) + \bar{a}\,\bar{b}\,f(0, 0, c...)$ y repitiendo el proceso se obtiene finalmente

$$f(a, b, c...) = (abc...)\,f(1, 1, 1...) + ... + (\bar{a}\,\bar{b}\,\bar{c}...)\,f(0, 0, 0...) \qquad [1]$$

Esta expresión indica que una función es igual a la suma de todos los productos canónicos afectados de un coeficiente igual al valor que toma la función al sustituir cada variable por 1 o 0 según en el producto canónico figure en forma directa o inversa respectivamente.

De igual forma se deduce que la expresión en forma de producto de sumas canónicas es:

$$f(a,b,c...) = (a + b + c + ... + f(0,0,0...)) ... (\bar{a} + \bar{b} + \bar{c} + ... + f(1,1,1...)) \qquad [2]$$

De la expresión [1] se deduce que toda función se puede representar mediante la suma de todos los productos canónicos multiplicados por un coeficiente igual a 1 si el término forma parte de la función e igual a 0 si no forma parte de ella.

Igualmente de la expresión [2] se deduce que la expresión canónica de sumas canónicas de una función es igual al producto de todas las sumas canónicas posibles, sumando a cada una de ellas un coeficiente igual a cero, si el término forma parte de la función e igual a 1 si no forma parte de ella.

Siendo 2^n el número de términos canónicos, el número de funciones canónicas de n variables es igual al de variaciones con repetición de dos elementos, 0 y 1, tomados de 2^n en 2^n, es decir 2^{2^n}.

Utilizando la notación numérica anteriormente indicada para expresar los términos canónicos, ambas ecuaciones [1] y [2] se pueden representar de la forma siguiente:

$$f(a,\ b,\ c,...) = \sum_{i=0}^{2^n-1} f(i)i = \prod_{i=0}^{2^n-1} [f(2^n - 1 - i) + i]$$

Estas dos expresiones numéricas permiten pasar con gran facilidad de una forma canónica a la otra, lo cual aplicaremos al tratar la simplificación de circuitos combinacionales.

En efecto, si un producto canónico i existe en una función debido a que su coeficiente es igual a 1, no existirá en la expresión en forma de productos de sumas canónicas de dicha función el término $2^n - 1 - i$ por ser su coeficiente también igual a la unidad. Por lo tanto si se tiene la expresión canónica en forma suma de productos, la expresión canónica en forma de producto de sumas se obtiene mediante el complemento a $2^n - 1$ de los productos canónicos que no forman parte de la función.

Por ejemplo, si $f = \sum_3 (0, 2, 5)$ tendremos:

$$f = \underset{3}{\amalg}(0,1,3,4,6)$$

Esto se demuestra de otra forma muy fácilmente aplicando los teoremas de De Morgan. Denominando al producto canónico por P con un subíndice igual al número decimal que le corresponde y a la suma canónica con una S y el subíndice respectivo, se verifica:

$$\bar{P}_i = S_{2^n-1-i}$$

Por ejemplo:

$$\overline{\bar{a}b\bar{c}} = \bar{P}_2 = a + \bar{b} + c = S_5$$

De igual forma $S_i = \bar{P}_{2^n-1-i}$

La función inversa de f está formada por los términos canónicos que no pertenecen a f. Invirtiendo la función f se obtiene de nuevo f, pero expresada en forma dual.

Por ejemplo si

$$f = \underset{3}{\Sigma}(0,3,7)$$

Se verifica

$$\bar{f} = \underset{3}{\Sigma}(1,2,4,5,6)$$

$$\bar{\bar{f}} = f = \overline{\underset{3}{\Sigma}(1,2,4,5,6)} = \underset{3}{\amalg}(1,2,3,5,6)$$

Cuando una función lógica se presenta de una forma no canónica su transformación en canónica resulta muy sencilla por procedimientos algebraicos.

Si se desea obtener la expresión canónica en forma de suma de productos canónicos, se operará algebraicamente aplicando las propiedades distributivas del producto con respecto a la suma, hasta obtener una expresión de suma de productos no canónicos. Para convertir cada uno de estos productos en canónicos se le multiplica por la suma de las variables que faltan en él y sus inversas.

Un ejemplo aclarará el procedimiento.

Sea la función $f = a(\bar{b} + \bar{c}) + c$

Aplicando la propiedad distributiva del producto con respecto a la suma, resulta:

$$f = a\bar{b} + a\bar{c} + c$$

De acuerdo con lo explicado anteriormente

$$f = a\bar{b}(c + \bar{c}) + a\bar{c}(b + \bar{b}) + c(a + \bar{a}) \cdot (b + \bar{b})$$

Y aplicando de nuevo la propiedad distributiva del producto con respecto a la suma, resulta:

$$f = a\bar{b}c + a\bar{b}\bar{c} + ab\bar{c} + a\bar{b}\bar{c} + abc + \bar{a}bc + a\bar{b}c + \bar{a}\bar{b}c$$

Suprimiendo los términos repetidos de acuerdo con el teorema 4 resulta:

$$f = a\bar{b}c + a\bar{b}\bar{c} + ab\bar{c} + abc + \bar{a}bc + \bar{a}\bar{b}c$$

La función f se puede expresar en la forma numérica abreviada antes indicada:

$$f = \sum_{a}(1,3,4,5,6,7)$$

De igual forma, si se desea obtener la expresión canónica en forma de producto de sumas canónicas, se operará algebraicamente aplicando la propiedad distributiva de la suma con respecto al producto hasta obtener una expresión de producto de sumas no canónicas. Para convertir cada una de estas sumas en canónica se le suma el producto de cada variable que falta en ella por su inversa. Un ejemplo aclarará el procedimiento.

Utilizaremos la misma función que en el caso anterior.

$$f = a(\bar{b} + \bar{c}) + c$$

Aplicamos la propiedad distributiva de la suma con respecto al producto:

$$f = (a + c)(\bar{b} + \bar{c} + c) = a + c$$

De acuerdo con la regla antes expresada resulta:

$$f = a + c + b\bar{b}$$

Y aplicando de nuevo la propiedad distributiva de la suma con respecto al producto, obtenemos:

$$f = (a + b + c)(a + \bar{b} + c)$$

La función f puede también expresarse en forma numérica

$$f = \prod_{3}(5,7)$$

2.4 TABLA DE VERDAD DE UNA FUNCION LOGICA

La tabla de verdad de una función lógica es una forma de representación de la misma, en la que se indica el valor 1 o 0 que toma la función para cada una de las combinaciones posibles de las variables de las cuales depende. En la tabla 2.2 se representa la tabla de verdad de una función de tres variables.

	c	b	a	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

TABLA 2.2

La deducción de la forma canónica de la función por medio de la tabla de verdad resulta sencilla.

Si, para una determinada combinación de las entradas, la función toma el valor lógico 1, el producto canónico de todos los posibles 2^n, que vale 1 para dicha combinación, ha de formar parte de la función. La deducción del producto canónico correspondiente es inmediata asignando al estado 0 la variable inversa y al estado 1 la variable directa.

Por ejemplo, el producto canónico correspondiente a la combinación 100 es el $c\bar{b}\bar{a}$. Sólo $c\bar{b}\bar{a}$ vale 1 cuando $c = 1$ y $b = a = 0$.

De lo dicho se deduce que la forma canónica de la función cuya tabla de verdad es la indicada en la tabla 2.2 es:

$$f = a\bar{b}\bar{c} + ab\bar{c} + \bar{a}\bar{b}c + \bar{a}bc + abc$$

Asignando a cada combinación binaria de entrada el número decimal equivalente se obtiene la expresión abreviada de la función:

$$f = \sum_3 (1,3,4,6,7)$$

Por el método indicado anteriormente se deduce la expresión canónica en forma de producto de sumas

$$f = \prod_3 (2,5,7)$$

La forma algebraica de la expresión canónica producto de sumas canónicas se puede también obtener directamente de la tabla de verdad observando las combinaciones para las cuales la función f toma el valor 0 y sustituyendo para cada una de ellas el valor 0 de una variable por su expresión directa y el valor 1 por su expresión inversa. Por ejemplo, en la función que ahora estudiamos correspondiente a la tabla 2.2, por ser $f = 0$ para la combinación $c = b = a = 0$ la suma canónica $a + b + c$ forma parte de la función. Por tanto, la expresión de la misma será:

$$f = (a + b + c)(\bar{a} + b + \bar{c})(a + \bar{b} + c)$$

que, como podemos comprobar, coincide con la expresión numérica antes indicada.

2.5 FUNCIONES IMPORTANTES DE UN ALGEBRA DE BOOLE

En apartados anteriores de este capítulo hemos estudiado, en primer lugar, las funciones básicas de un álgebra de Boole producto lógico, suma lógica, e inversión y posteriormente las funciones NO-O (NOR) y NO-Y (NAND).

Además de estas funciones existen otras cuya importancia se deriva de que pueden ser utilizadas para la realización de las demás funciones lógicas.

2.5.1 Función O-exclusiva

La función O-exclusiva de dos variables a y b es aquella que toma el valor

b	a	f_a
0	0	0
0	1	1
1	0	1
1	1	0

TABLA 2.3

uno cuando una de las variables toma el valor uno y la otra el valor cero o viceversa. La función O-exclusiva se representa mediante el símbolo $\oplus$.

En la tabla 2.3 se representa la tabla de verdad de la función O-exclusiva de dos variables:

$$f_0 = a \oplus b$$

En las figuras 2.10 y 2.11 se representan respectivamente los símbolos antiguo y nuevo normalizado de esta función.

FIGURA 2.10.—**Símbolo lógico no normalizado de la función O-exclusiva.**

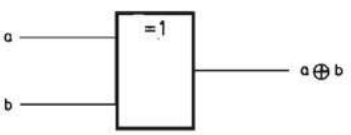


FIGURA 2.11.—**Símbolo lógico normalizado de la función O-exclusiva.**

De la tabla 2.3 se deducen las expresiones canónicas de la función O-exclusiva:

$$f_0 = a\bar{b} + \bar{a}b = (a + b)(\bar{a} + \bar{b})$$

que no son simplificables.

La función O-exclusiva se puede realizar con puertas NO-Y o NO-O, pero el perfeccionamiento de la tecnología de fabricación de los circuitos ha permitido fabricar bloques funcionales en circuito integrado que realizan esta función.

Las propiedades de la función O-exclusiva de n variables se deducen aplicándola primero a dos variables, seguidamente al resultado obtenido y una tercera variable y así sucesivamente.

Se comprueba fácilmente que la función O-exclusiva de n variables toma el valor lógico uno si se encuentra un número impar de ellas en estado uno, y el valor lógico cero si es un número par de ellas el que posee el valor lógico uno:

$$f_0 = a \oplus b \oplus c \oplus d \ldots \oplus n$$

$f_0 = 1$ si un número impar de variables está en uno
$f_0 = 0$ si un número par de variables está en uno
(se considera el cero un número par)

La función O-exclusiva presenta las propiedades siguientes, que el lector puede demostrar a partir de los postulados y teoremas estudiados en los apartados anteriores de este capítulo:

$$f_0 = a \oplus b = \overline{a \oplus \bar{b}} = \overline{\bar{a} \oplus b} = \bar{a} \oplus \bar{b}$$

2.5.2 Función equivalencia o comparación

La función equivalencia de dos o más variables es aquella que toma el valor uno cuando todas las variables de entrada se encuentran en el mismo estado lógico y el valor cero en caso contrario.

La tabla de verdad de la función equivalencia de dos variables se representa en la tabla 2.4. Se observa que esta función es el inverso de la función O-exclusiva de dos variables.

$$f_e = \overline{a \oplus b}$$

Las expresiones canónicas de la función equivalencia se deducen de la tabla 2.4.

b	a	f_e
0	0	1
0	1	0
1	0	0
1	1	1

TABLA 2.4

Aplicando la propiedad de la función O-exclusiva estudiada en el apartado anterior resulta:

$$f_e = \overline{a \oplus b} = a \oplus \bar{b} = \bar{a} \oplus b$$

y por tanto la realización física más sencilla de esta función se logra mediante una puerta O-exclusiva de dos entradas a la cual se le aplica una variable en forma directa y la otra invertida tal como se representa en la figura 2.12 con el símbolo antiguo y en la figura 2.13 con el símbolo nuevo. En el nuevo sistema de representación se ha reservado el indicativo « = » para la función equivalencia que, por tanto, puede representarse mediante el símbolo de la figura 2.14.

FIGURA 2.12.—**Símbolo lógico no normalizado de la función equivalencia.**

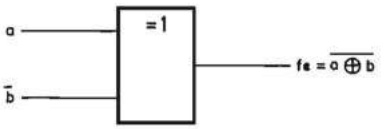


FIGURA 2.13.—**Representación de la función equivalencia utilizando el símbolo lógico normalizado de la función O-exclusiva.**

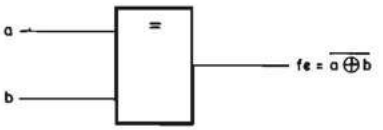


FIGURA 2.14.—**Símbolo lógico normalizado de la función equivalencia.**

PROBLEMAS

1. *a*) Hallar las expresiones canónicas suma de productos y producto de sumas de las funciones:

$$f_1 = ab + a\bar{b}c + \bar{a}\bar{b}$$
$$f_2 = abc\bar{d} + ab\bar{c} + \bar{a}bd$$

b) Representar la tabla de verdad de las dos funciones anteriores.

2. Dada la función f_1 representada mediante la expresión canónica de suma de productos:

$$f_1 = \sum_4 (0, 1, 2, 3, 12, 15)$$

a) Obtener la expresión canónica de producto de sumas.
b) Obtener las dos expresiones canónicas algebraicas de esta función.
c) Representar la tabla de verdad de esta función.

3. La función $f(a, b, c, d)$ cumple la siguiente tabla de verdad.

d	*c*	*b*	*a*	*f*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

a) Obtener las expresiones numéricas canónicas de suma de productos y de productos de sumas.
b) Obtener las expresiones algebraicas canónicas de esta función.
c) Simplificar las expresiones anteriores por procedimientos algebraicos.

4. Una función de tres variables $f(a, b, c)$ ha de tomar el valor cero cuando la variable b se encuentre en estado uno y la variable a no esté en estado uno. En los demás casos posibles ha de estar en estado uno.

a) Realizar la tabla de verdad de esta función.
b) Obtener las formas canónicas de suma de productos y producto de sumas.

5. Realizar la tabla de verdad de un convertidor de código BCD exceso tres a BCD natural. Indicar mediante el símbolo X el valor de las variables correspondientes a aquellas combinaciones de entrada que no pueden existir.

6. Demostrar las siguientes propiedades de la función O-exclusiva:
a) $a \oplus b = \bar{a} \oplus \bar{b}$
b) $a(b \oplus c) = ab \oplus ac$.

7. Dada la función:

$$f = \overline{a + b} + \bar{a}bc + \overline{a(b + c)}$$

a) Obtener las expresiones canónicas en forma de producto de sumas y suma de productos.
b) Representar la tabla de verdad de esta función.

Capítulo

3

Sistemas combinacionales

3.1 GENERALIDADES

Sistemas lógicos combinacionales son aquellos en los que en cada instante, el estado lógico de sus salidas depende únicamente del estado de sus entradas. Por lo tanto, en ellos no es necesario tener en cuenta la noción de tiempo. Si analizamos esta definición, observamos que un sistema combinacional es realmente una función lógica tal como se ha definido en el capítulo anterior. Por tanto, los sistemas combinacionales pueden ser representados mediante una tabla de verdad o mediante las expresiones numéricas estudiadas en el capítulo anterior correspondiente a una suma de productos canónicos y un producto de sumas canónicas.

En el capítulo anterior se ha estudiado tambien que las funciones lógicas podían simplificarse por métodos algebraicos aplicando los postulados y teoremas de álgebra de Boole. En la práctica, en especial si la función depende de más de tres variables, los métodos algebraicos, aunque son utilizables, no representan una forma sistemática de minimizar las funciones lógicas. Por ello en este capítulo se desarrollan diversos métodos sistemáticos de minimizar las funciones lógicas.

El diseño de todo sistema combinacional se inicia mediante la obtención de una tabla de verdad a partir de las especificaciones que indican los valores que debe tomar la función para cada una de las combinaciones de las variables de entrada de las cuales depende. De la tabla de verdad se deducen las expresiones canónicas a partir de las cuales se realizará la simplificación. Un ejemplo ayudará a aclarar el procedimiento.

Ejemplo 3.1: Diseñar un sistema combinacional de tres variables de entrada a, b y c cuya salida debe tomar el valor uno solamente cuando dos variables de entrada toman el valor uno.

En primer lugar se realiza la tabla de verdad de esta función, para lo cual se indican en orden creciente en el sistema binario natural todas las combinaciones posibles de las variables de entrada a, b y c. (tabla 3.1)

Seguidamente y en una nueva columna se indica el valor que toma la salida f para cada combinación de entrada, de acuerdo con las especificaciones del enun-

ciado. Para cada combinación de entrada en que el número de unos sea distinto de dos, la función ha de tomar el valor cero y, por el contrario, cuando dicho número sea igual a dos, la función f ha de tomar el valor uno. En la tabla 3.1 se representa la tabla de verdad obtenida.

A partir de la tabla de verdad se obtienen las expresiones canónicas por el procedimiento indicado en el apartado 2.4.

Dichas expresiones son:

$$f = \sum_{3} (3,5,6)$$

$$f = \prod_{3} (0,3,5,6,7)$$

En sucesivos apartados se completará el diseño de este sistema combinacional.

c	b	a	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

TABLA 3.1

3.2 SIMPLIFICACION DE LAS FUNCIONES LOGICAS

Los criterios de minimización de la expresión de una función lógica han tenido una gran importancia cuando sólo existían circuitos de pequeña escala de integración. Posteriormente la aparición de los circuitos de escala de integración media disminuyó su interés para el diseñador de sistemas electrónicos. El progreso de la microelectrónica con el desarrollo de los circuitos combinacionales programables (que se estudian en el apartado 3.8 al final de este capítulo) ha hecho que los métodos de minimización se lleven a cabo mediante programas de computador, incorporándolos así a las técnicas de diseño microelectrónico asistido por computador.

El criterio de minimización más utilizado es el de obtener una expresión en forma de suma de productos o producto de sumas que tenga un número mínimo de términos con el menor número de variables posible en cada uno de ellos.

Para obtener una expresión mínima de suma de productos o producto de sumas partiremos de la forma canónica correspondiente. En general se han de obtener ambas expresiones y utilizar la más sencilla de ellas para realizar prácticamente la función.

En los métodos que estudiaremos en los apartados siguientes aplicaremos de una forma sistemática y adecuada la siguiente propiedad del álgebra de Boole:

$$abc... + \bar{a}bc... = bc... \qquad [1]$$
$$(a + b + c...)(\bar{a} + b + c + ...) = (b + c + ...)$$

La demostración de estas ecuaciones es inmediata aplicando los postulados del álgebra de Boole indicados en el capítulo anterior. La primera de las dos expresiones indican que la suma de dos productos canónicos adyacentes lógicamente, es decir que difieren solamente por el estado de una de las variables, se reduce a un único producto en el cual se ha suprimido dicha variable. La segunda expresión es la dual de la primera y por lo tanto indica lo mismo para las sumas canónicas.

Esta propiedad algebraica de simplificación de las funciones lógicas se puede indicar también de una forma numérica utilizando los equivalentes numéricos de las combinaciones binarias tal como se indicó en el capítulo anterior.

Supongamos una expresión suma de productos canónicos de cuatro variables:

$$abcd + abc\bar{d} + ab\bar{c}d + ab\bar{c}\bar{d}$$

De acuerdo con la ecuación [1] se tiene:

$$abcd + abc\bar{d} = abc(d+\bar{d}) = abc$$
$$\text{y } ab\bar{c}d + ab\bar{c}\bar{d} = ab\bar{c}(d+\bar{d}) = ab\bar{c}$$

Estas ecuaciones se pueden representar en forma numérica

$$\begin{array}{ccc} abcd + abc\bar{d} = abc \\ 15 \qquad 7 \rightarrow 7-15 \end{array}$$
$$\begin{array}{ccc} ab\bar{c}d + ab\bar{c}\bar{d} = ab\bar{c} \\ 11 \qquad 3 \rightarrow 3-11 \end{array}$$

Se observa que los números difieren en una potencia de dos igual al peso de la variable binaria que desaparece.

Los dos términos abc y $ab\bar{c}$ pueden a su vez agruparse:

$$\begin{array}{ccc} abc + ab\bar{c} = ab(c+\bar{c}) = ab \\ 7-15 \quad 3-11 \rightarrow 3-7-11-15 \end{array}$$

La expresión resultante es, pues:

$$ab = 3 - 7 - 11 - 15.$$

Otro ejemplo es:

$$\begin{array}{ccc} abc + ab\bar{c} = ab \\ 7 \qquad 3 \rightarrow 3-7 \end{array}$$
$$\begin{array}{ccc} a\bar{b}c + a\bar{b}\bar{c} = a\bar{b} \\ 5 \qquad 1 \rightarrow 1-5 \end{array}$$
$$\begin{array}{ccccc} a\bar{b} & + & ab & = & a \\ 1-5 & & 3-7 & \rightarrow & 1-3-5-7 \end{array}$$

Se observa que el número de términos canónicos que quedan comprendidos en un término reducido es una potencia de dos.

Aplicando estas propiedades se logra por lo tanto reducir al mínimo cualquier expresión lógica en forma de producto de sumas o de suma de productos. La expresión final, en la que no se puede suprimir ningún término ni eliminar variables de ellos, se denomina expresión irreducible.

Combinando los términos para su simplificación de diferentes maneras, se

obtienen a veces varias expresiones irreducibles. De todas ellas es necesario elegir la menos compleja para su realización.

3.2.1 Métodos tabulares de Karnaugh y Veitch de simplificación de las funciones lógicas

Aunque en cierto modo es sistemática la aplicación directa del método algebraico, no lo es totalmente porque, en general, existen diversas formas de agrupar los términos para su reducción y por tanto, varias expresiones irreducibles. Por ello se idearon los métodos tabulares que constituyen una forma gráfica de representar la tabla de verdad de una función lógica.

Se ha visto en el apartado anterior que los términos canónicos adyacentes pue-

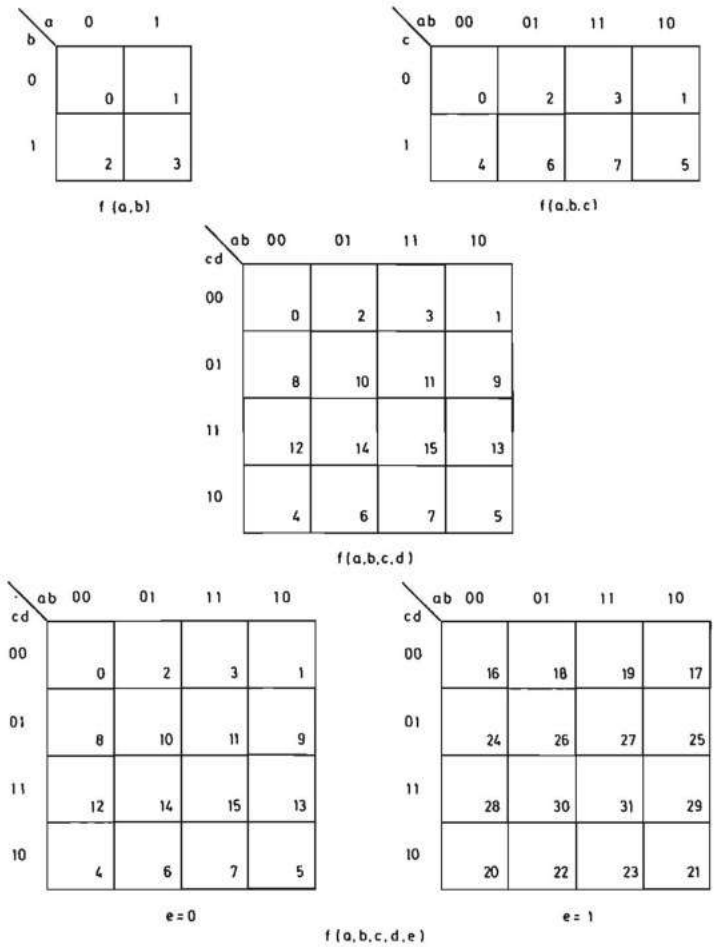


FIGURA 3.1.—**Tablas de Karnaugh.**

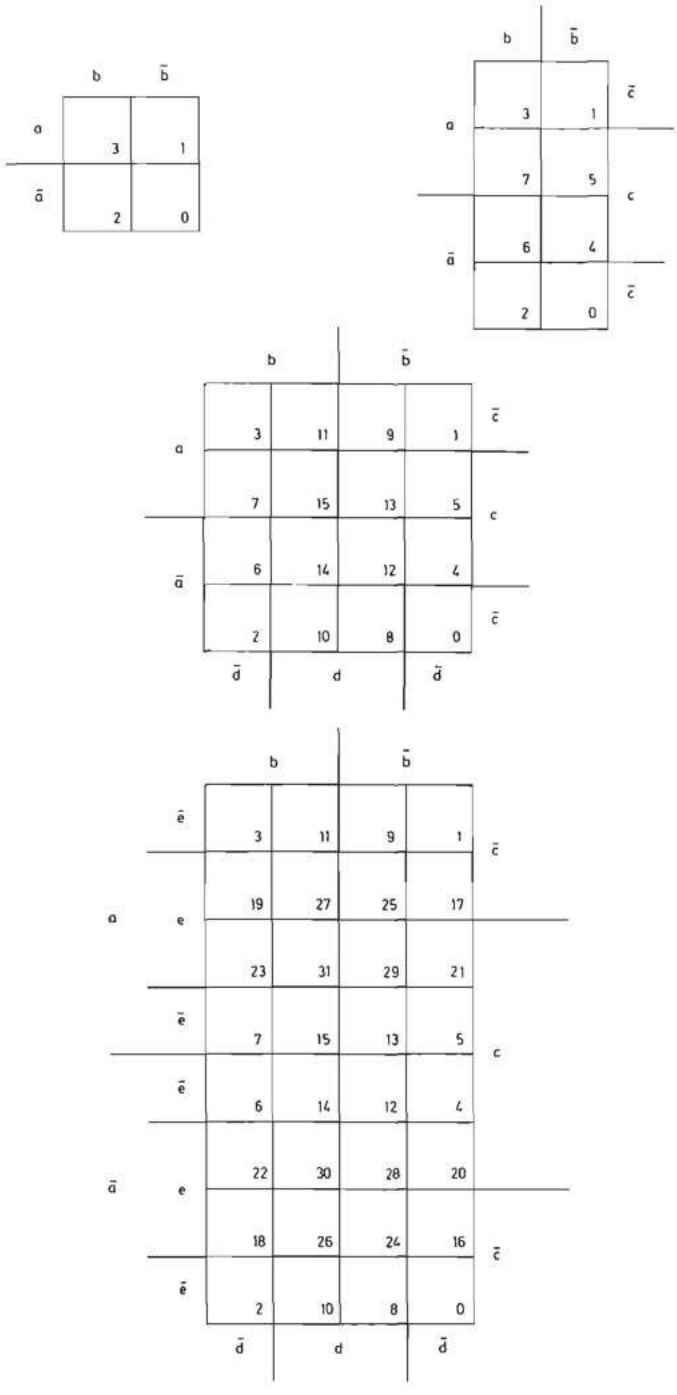


FIGURA 3.2.—**Tablas de Veitch.**

den reducirse a un solo término en el cual se ha suprimido la variable cuyo estado es diferente en ambos.

En los métodos tabulares los términos canónicos adyacentes se agrupan en una tabla de tal manera que estén físicamente contiguos y por tanto sea muy sencillo realizar las agrupaciones que permiten reducir al mínimo la expresión de la función.

Se han adoptado dos formas diferentes de realizar las tablas, que reciben los nombres de los primeros matemáticos que las realizaron, Karnaugh y Veitch.

En la figura 3.1 se representan las tablas de Karnaugh de funciones de dos a cinco variables y en la figura 3.2 las de Veitch de igual número de variables.

Tal como se observa en ambas figuras, cada cuadrado corresponde a un término (producto o suma) canónico cuyo número se indica en el vértice inferior. Los cuadros que tienen un lado común, es decir, que son físicamente adyacentes, corresponden a términos canónicos que son lógicamente adyacentes y los números decimales que les corresponden se diferencian en una potencia de dos. Además, los cuadrados de la fila superior son adyacentes a los respectivos de la fila inferior y los de la columna de la izquierda a los de la derecha.

En la tabla de Karnaugh de cinco variables, los de la tabla de la izquierda son adyacentes a los correspondientes de la tabla de la derecha (p. ej., el 0 y el 16) y en la tabla de Veitch de cinco variables existe un eje de adyacencia horizontal que divide a la tabla en dos partes cuyos cuadrados simétricos son adyacentes dos a dos.

En este estudio se utilizan exclusivamente las tablas de Karnaugh. Los cuadrados correspondientes a los términos canónicos que forman parte de la función se indican mediante un uno y los correspondientes a los términos que no forman parte de la función se dejan en blanco.

Para obtener la expresión algebraica más sencilla de la función es necesario realizar en la tabla el mínimo número de agrupaciones de términos de la máxima complejidad, de modo que cada uno cubra todos los unos de la tabla. De acuerdo con lo indicado en el apartado 3.2, el número de términos canónicos adyacentes que pueden agruparse es una potencia de dos.

El procedimiento sistemático de obtener la expresión más simple es el siguiente:

1. Se toman todos los «unos» que no se pueden combinar con ningún otro.
2. Se forman los grupos de dos «unos» que no pueden formar un grupo de cuatro.
3. Se forman los grupos de cuatro «unos» que no pueden formar un grupo de ocho.
4. Cuando se hayan cubierto todos los unos se detiene el proceso.

En la práctica es necesario realizar este proceso para ambas expresiones canónicas y elegir la más sencilla de las resultantes.

Varios ejemplos aclararán lo expuesto.

Ejemplo 3.2

Sea la función

$$f_1 = \sum_4 (2, 3, 5, 7, 10, 11, 15) = \prod_4 (1, 2, 3, 6, 7, 9, 11, 14, 15)$$

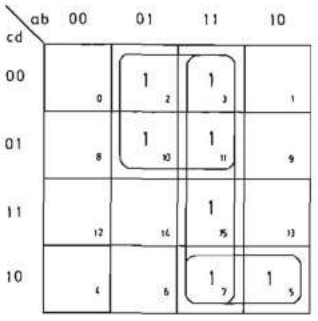


FIGURA 3.3.—**Tabla de Karnaugh de la expresión canónica de la función f_1 en forma de suma de productos.**

En la figura 3.3 se representa la tabla de Karnaugh de la expresión canónica de f_1 en forma de suma de productos. La expresión mínima es única y está compuesta por dos agrupaciones de cuatro términos y una de dos términos. El producto lógico que corresponde a cada grupo se obtiene eliminando las variables que toman el valor 0 en la mitad de las células y el valor 1 en la otra mitad y asignando la forma directa a la variable que toma el valor 1 y la forma inversa a la que toma el valor 0, de acuerdo con el convenio indicado en el capítulo 2. Por ejemplo, al agrupamiento formado por las casillas 2, 3, 10 y 11 le corresponde el producto $b\bar{c}$.

La expresión algebraica resultante de f_1 es, pues:

$$f_1 = b\bar{c} + ab + ac\bar{d}$$

Simplificando la expresión canónica en forma de producto de sumas cuya tabla de Karnaugh se representa en la figura 3.4 se tiene:

$$f_1 = (a + \bar{c})(b + c)(b + \bar{d})$$

Se observa que esta segunda expresión es más sencilla que la primera porque contiene el mismo número de términos y una variable menos en uno de ellos. Por tanto deberá ser utilizada para realizar en la práctica la función.

Al realizar la minimización de una función puede suceder que exista más de

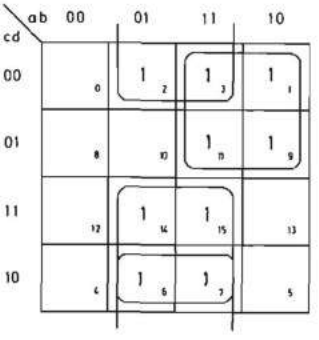


FIGURA 3.4.—**Tabla de Karnaugh de la expresión canónica de la función f_1 en forma de producto de sumas.**

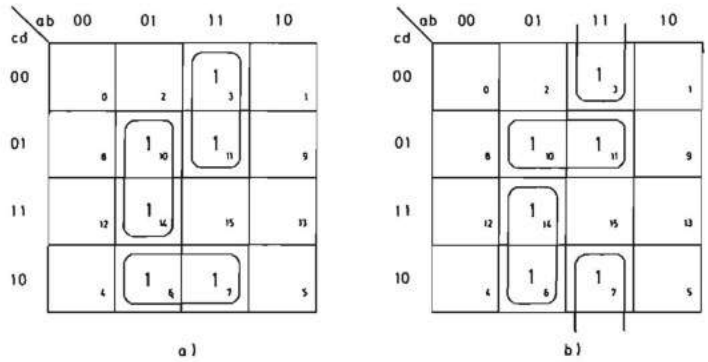


FIGURA 3.5.—**Tablas de Karnaugh de la expresión canónica de la función f_2 en forma de producto de sumas.**

una forma mínima irreducible. En este caso se puede elegir cualquiera de las dos para realizar la función tal como se indica en el ejemplo siguiente.

Ejemplo 3.3: Sea la función:

$$f_2 = \sum_4 (3, 6, 7, 10, 11, 14)$$

En la figura 3.5 se presentan dos tablas de Karnaugh de esta función correspondientes a las dos formas posibles de agrupar los términos canónicos. Por ejemplo el término 3 se puede agrupar con el 11 o con el 7. Las dos expresiones mínimas que corresponden respectivamente a las tablas *a* y *b* de la figura 3.5 son:

$$f_2 = ab\bar{c} + \bar{a}bd + bc\bar{d}$$
$$f_2 = ab\bar{d} + b\bar{c}d + \bar{a}bc$$

Ejemplo 3.4: Se continúa el diseño del sistema del ejemplo 3.1, simplificando las expresiones canónicas obtenidas, que fueron las siguientes:

$$f = \sum_3 (3, 5, 6) = \prod_3 (0, 3, 5, 6, 7)$$

Las tablas de Karnaugh de ambas expresiones se presentan en las figuras 3.6 y 3.7.

En la tabla de la figura 3.6 de la suma de productos canónicos se observa que no existen términos adyacentes y por tanto la expresión mínima es equivalente a la canónica:

$$f = ab\bar{c} + \bar{a}bc + a\bar{b}c$$

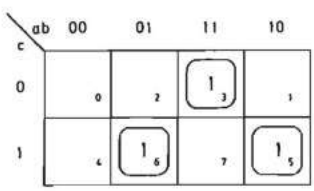


FIGURA 3.6.—**Tabla de Karnaugh de la función $f = \sum_3 (3, 5, 6)$.**

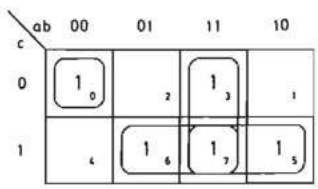


FIGURA 3.7.—**Tabla de Karnaugh de la función $f = \prod_3 (0, 3, 5, 6, 7)$.**

En la tabla de la figura 3.7 del producto de sumas canónicas se observa que la suma canónica 3 se puede agrupar con la 7. De igual forma la 6 y la 5 se pueden agrupar con la 7. No existe ninguna suma canónica adyacente a la 0 que tome el valor lógico uno y por tanto ésta se ha de realizar por sí sola. La expresión resultante obtenida es:

$$f = (\bar{a} + \bar{b} + \bar{c})(a + b)(b + c)(a + c)$$

La aplicación del método tabular de Karnaugh, tal como se acaba de exponer, a las funciones de más de cinco variables no resulta realizable en la práctica porque es difícil observar todos los términos adyacentes. Por ello tiene gran interés el método que se expone seguidamente, que permite tratar funciones de $n + 1$, $n + 2$, etc. variables con una tabla de Karnaugh de n variables.

En el procedimiento que se acaba de estudiar se coloca un uno en los cuadrados de la tabla de Karnaugh de suma de productos correspondientes a aquellos productos canónicos que forman parte de la función, y un cero en todos los demás.

Con una tabla de n variables es posible representar una función de $n + 1$ asignando a cada cuadrado dos términos canónicos en lugar de uno solo. De esta forma cada cuadrado puede contener un uno, un cero, una de las $n + 1$ variables en forma directa o en forma inversa. La elección de la variable es arbitraria.

La mejor manera de desarrollar lo que se acaba de exponer es mediante un ejemplo.

Ejemplo 3.5:

Sea la función de cinco variables:

$$f(a, b, c, d, e) = \sum (0, 1, 2, 3, 8, 9, 10, 16, 17, 18, 19, 24, 25, 27)$$

Esta función se puede simplificar mediante el procedimiento general realizando las tablas indicadas en la figura 3.8*a* y teniendo en cuenta que los términos simétricos de ambas tablas son adyacentes porque se diferencian solamente en el estado de la variable *e*.

Pero se puede realizar más fácilmente la simplificación si ambas tablas se reúnen en una sola asignando a cada cuadrado dos términos canónicos. Se obtiene así la tabla de la figura 3.8*b*. A todos aquellos cuadrados que tienen un uno en ambas tablas de la figura 3.8*a* se les asigna un uno en la tabla de la figura 3.8*b*. A aquellos cuadrados que poseen un uno en una tabla y un cero en la otra se les asigna el estado de la variable *e* correspondiente a la tabla del cuadrado al que le corresponde el uno. Por ejemplo, al cuadrado 10 le corresponde un uno y al 26 un cero en las tablas de la figura 3.8*a*, y por ello al cuadrado 10-26 de la figura 3.8*b* se le asigna la variable $\bar{e}$. Por la misma razón al cuadrado 11-27 se le asigna la variable *e*.

El agrupamiento de términos en la tabla de la figura 3.8*b* se realiza de la forma siguiente:

a) Se agrupan entre sí los términos que poseen la variable *e* y con los que poseen uno, de tal manera que se cubran de la forma más sencilla posible todos los cuadra-

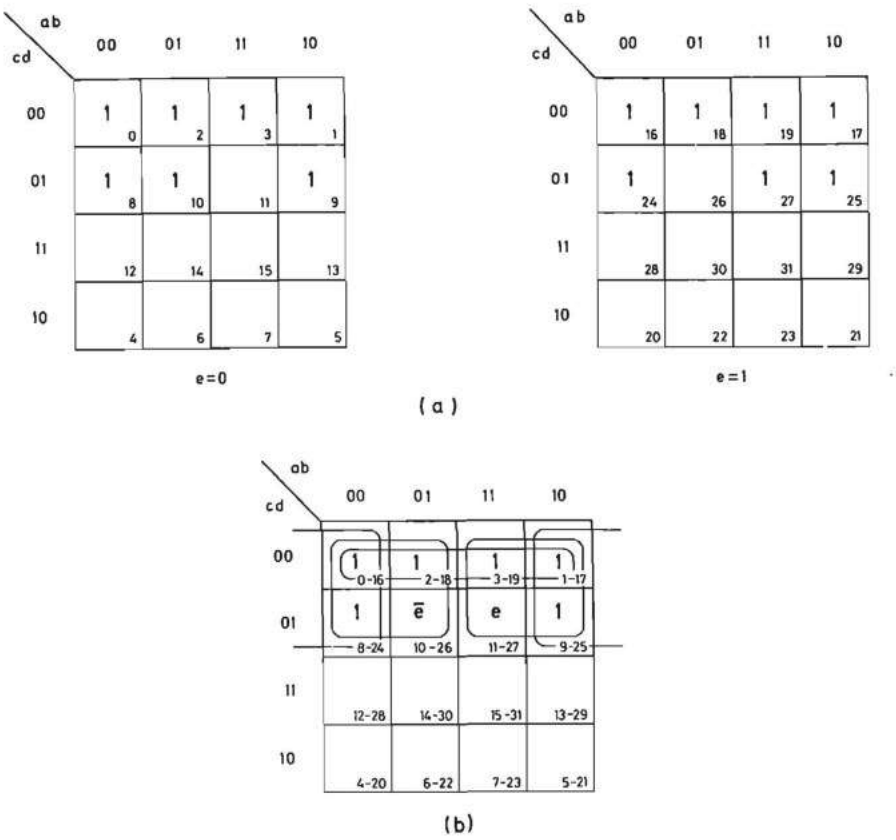


FIGURA 3.8.—**Tablas de Karnaugh de la función** $f(a,b,c,d,e) = \sum_{3}(0, 1, 2, 3, 8, 9, 10, 16, 17, 18, 19, 24, 25, 27)$: *a*) **Tablas independientes para** $e = 0$ **y** $e = 1$; *b*) **Tabla única.**

dos que poseen la variable *e*. En la expresión del término obtenido aparece la variable *e* en forma directa.

Se obtiene así el término:

$$3 - 19 - 1 - 17 - 11 - 27 - 9 - 25 \equiv a\bar{c}e$$

b) Se agrupan entre sí todos los cuadros que poseen $\bar{e}$ y con los que poseen uno, de tal manera que se cubran de la forma más sencilla posible todos los cuadrados que poseen la variable $\bar{e}$. En la expresión del término obtenido aparece la variable *e* en forma inversa.

Se obtiene así el término:

$$0 - 16 - 2 - 18 - 8 - 24 - 10 - 26 \equiv \bar{a}\bar{c}\bar{e}$$

c) Se agrupan de la forma más sencilla posible todos los términos que poseen un uno y que no han sido realizados conjuntamente con *e* y con $\bar{e}$ en los pasos anteriores.

Como en la figura 3.8*b* no existe ningún uno realizado conjuntamente con e y con $\bar{e}$ se unen entre sí los unos de la fila superior y se obtiene así el término:

$$0-16-2-18-3-19-1-17 \equiv \bar{c}\bar{d}$$

De igual forma se obtiene el término:

$$0-16-1-17-8-24-9-25 \equiv \bar{b}\bar{c}$$

Por lo tanto, la expresión resultante de la función f es:

$$f = \bar{c}d + \bar{b}\bar{c} + a\bar{c}e + \bar{a}\bar{c}\bar{e}$$

Si, por el contrario, a la función f le corresponde la tabla de Karnaugh de la figura 3.9 su expresión numérica es:

$$f = ebd + \bar{e}ad$$

Los términos que valen uno no hay que realizarlos de forma independiente porque ya están comprendidos tanto en el término que cubre la variable e como en el que cubre la variable $\bar{e}$.

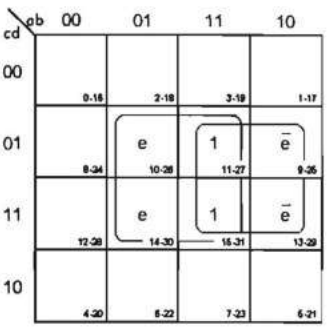


FIGURA 3.9

Para facilitar la comprensión del procedimiento se realizaron las tablas de la figura 3.8*a*, pero en la práctica se obtiene la tabla de la figura 3.8*b* directamente.

En una tabla de Karnaugh de cuatro variables se pueden representar funciones de 6 y más variables, en especial en el caso de que las funciones no estén definidas para todas las combinaciones posibles de las variables de entrada tal como se explica en el apartado 3.3.

3.2.2 Método numérico de Quine-McCluskey de simplificación de los sistemas lógicos combinacionales

Los métodos tabulares estudiados anteriormente son de aplicación práctica para simplificar funciones de un máximo de cuatro a cinco variables, pero, cuando dicho

número es mayor y también cuando se trata de una multifunción cuya simplificación se estudiará en un apartado posterior, es necesario recurrir a otros métodos, de los cuales el numérico es el de uso más corriente.

Por otra parte, el método numérico es adecuado para ser realizado mediante un algoritmo ejecutable por un programa de computador. Resulta, por lo tanto, un método idóneo en las técnicas de diseño asistido por computador.

Tabla 1

N.° de unos		
0	0	x
	1	x
1	4	x
	8	x
	3	x
2	12	x
	7	x
3	11	x
	13	x
4	15	x

Tabla 2

		Dif.
0-1		1
0-4	x	4
0-8	x	8
1-3		2
4-12	x	8
8-12	x	4
3-7	x	4
3-11	x	8
12-13		1
7-15	x	8
11-15	x	4
13-15		2

Tabla 3

	Dif.
0-4-8-12	(4,8)
0-8-4-12	(8,4)
3-7-11-15	(4,8)
3-11-7-15	(8,4)

Tabla de Quine-McCluskey de la función

$f(a, b, c, d) = \sum_4(0, 1, 3, 4, 7, 8, 11, 12, 13, 15)$

TABLA 3.2

Este método está basado en el convenio estudiado en el capítulo 2 de asignar un número decimal a cada término canónico. Si dos términos canónicos difieren en una sola variable, se ha visto anteriormente que sus números correspondientes difieren en una potencia de dos y que pueden agruparse en un solo término que ya no es canónico y al cual le falta dicha variable. A su vez, si dos términos a los cuales les falte la misma variable difieren en una misma potencia de dos, pueden ser agrupados en un nuevo término al cual le falte la variable correspondiente a dicha diferencia.

Repitiendo este proceso se logra obtener todos los términos primos, que son aquellos que contienen el máximo número de términos canónicos de la función y que, por otra parte, no existe ningún término de menor complejidad que los contenga.

En la tabla 3.2 se representan las tablas numéricas de la función de cuatro variables $f(a, b, c, d) = \sum_4(0, 1, 3, 4, 7, 8, 11, 12, 13, 15)$.

Primero se forma una tabla en la que los términos canónicos se ordenan en grupos de acuerdo con el número de unos que posee la combinación binaria que les corresponde. De esta forma, para realizar las agrupaciones es necesario comparar solamente los términos de cada grupo con los del siguiente.

Partiendo de esta primera tabla se forma una segunda comparando los términos canónicos que pertenecen a grupos adyacentes y agrupando en un solo término

aquellos cuya diferencia (se toma como minuendo el término que contiene mayor número de unos) sea una potencia de dos positiva. Por ejemplo, en la función de cuatro variables representada en la tabla 3.2 se pueden agrupar los términos canónicos 4 y 12 porque $12 - 4 = 8 = 2^3$ y además se puede comprobar que sus equivalentes binarios respectivos 0100 y 1100 son adyacentes. Por el contrario, 4 y 3 no se pueden agrupar porque $3 - 4 = -1$ y, en efecto, sus equivalentes binarios 0100 y 0011 no son adyacentes. Todos los términos de la primera tabla que han sido utilizados para realizar la segunda se marcan con una cruz, lo cual indica que no son términos primos.

En esta segunda tabla existe una columna en la cual se indica la diferencia entre los términos canónicos que forman parte de cada elemento de la misma. A partir de ella se forma una tercera tabla agrupando los términos pertenecientes a grupos adyacentes cuya diferencia es igual y que además difieren entre sí en una potencia de dos. Por ejemplo, los términos $0 - 4$ y $8 - 12$ de la segunda tabla de la tabla 3.2 se pueden agrupar entre sí porque su diferencia es la misma (4) y además difieren en una potencia de dos positiva ($8 - 0 = 12 - 4 = 8$).

En la tercera tabla se indican en una segunda columna ambas diferencias, la interna de cada grupo y la que existe entre los grupos que se unen. Por ejemplo, la diferencia del grupo $0 - 4 - 8 - 12$ es 4,8. El proceso se continúa realizando tablas sucesivas hasta que no es posible realizar más agrupaciones.

Debido a que una expresión formada por el agrupamiento de cuatro términos canónicos adyacentes puede obtenerse de dos formas diferentes, tal como se indica en la tabla de Karnaugh de la figura 3.10 se obtienen en esta tabla todos los términos por duplicado con diferente ordenación. De ellos solamente es necesario considerar uno, por ejemplo, aquel en que los términos están ordenados en orden creciente.

La cuarta tabla, en caso de que sea posible, se forma agrupando los términos de grupos adyacentes de la tercera tabla cuyas diferencias coinciden y que además difieren en una potencia de dos.

Una vez obtenidas todas las tablas se tienen todos los términos primos que pueden utilizarse para realizar la función, que son aquellos que no han sido marcados con una cruz.

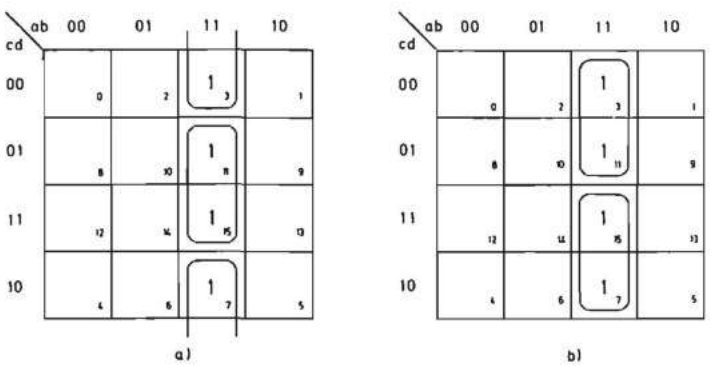


FIGURA 3.10.—**Diferentes formas de agrupar los términos adyacentes en una tabla de Karnaugh.**

Términos primos		0	1	3	4	7	8	11	12	13	15
A	0 - 1	X	X								
B	1 - 3		X	X							
C	12 - 13								X	X	
D	13 - 15									X	X
E	0 - 4 - 8 - 12	X			⊗		⊗		X		
F	3 - 7 - 11 - 15			X		⊗		⊗			X
					↑	↑	↑	↑			

TABLA 3.3.—**Tabla de los términos primos de la función** $f = \sum_{d} (0, 1, 3, 4, 7, 8, 11, 12, 13, 15)$.

En el ejemplo de la tabla 3.2 los términos primos son:

0-1, 1-3, 12-13, 13-15, 0-4-8-12 y 3-7-11-15.

Ahora es necesario elegir la mínima combinación de estos términos primos que cubra la función. Para ello se puede utilizar una representación como la indicada en la tabla 3.3 para la función que hemos estudiado en párrafos anteriores. Esta tabla contiene una columna por cada término canónico que forma parte de la función y una fila por cada término primo.

En la fila correspondiente a un determinado término primo, se coloca el símbolo X en las columnas cuyo número está contenido en el término.

En primer lugar se observa en la tabla si existe alguna columna que contenga un solo símbolo X, y se indican mediante una flecha. El término primo que realice esa X es esencial, es decir, ineludiblemente ha de formar parte de la función. Por tanto, todos los términos canónicos incluidos en ese término primo esencial quedan realizados por él. En la tabla 3.3, los términos 0-4-8-12 y 3-7-11-15 son esenciales y por tanto, al formar parte de la expresión final, quedan realizados los términos 0, 3, 4, 7, 8, 11, 12 y 15 que se indican en un recuadro.

Ahora es necesario elegir la combinación más sencilla de los términos primos restantes que realiza el resto de los términos canónicos.

Esta elección se puede hacer de dos formas diferentes. La primera es realizar una tabla reducida tal como se indica en la tabla 3.4 cuyas columnas son los

Términos primos no esenciales	1	13
0 - 1	X	
1 - 3	X	
12 - 13		X
13 - 15		X

TABLA 3.4.—**Tabla reducida**

términos canónicos que todavía no han sido realizados y cuyas filas corresponden a los términos primos no esenciales. Se observa que, para realizar el producto canónico 1, se pueden utilizar el término 0-1 o el 1-3 y para realizar el 13, se pueden utilizar el 12-13 o, el 13-15. Por lo tanto existen cuatro formas de combinar los términos primos para realizar el 1 y 13 que son:

0-1 y 12-13; 0-1 y 13-15; 1-3 y 12-13; 1-3 y 13-15.

La segunda forma de realizarlo, que es más sistemática, consiste en la obtención de una expresión algebraica igual al producto lógico de las sumas lógicas de los términos primos disponibles para realizar cada término canónico. Para facilitar la escritura, le asignaremos a cada término primo una letra (tabla 3.3).

Para realizar el término 1 en la tabla 3.3 se dispone del 0-1 o el 1-3 y para realizar el 13 se dispone de los términos 12-13 o 13-15.

Por lo tanto la expresión algebraica será

$$(A + B)\,(C + D)$$

y aplicando la propiedad distributiva del producto con respecto a la suma lógica se tiene:

$$(A + B)\,(C + D) = AC + AD + BC + BD$$

Por lo tanto, se puede coger cualquiera de las cuatro combinaciones A y C, A y D, B y C o B y D para realizar los términos 1 y 13 y el resultado es el mismo que el obtenido anteriormente.

Las expresiones más reducidas de f son las cuatro siguientes:

$$\begin{aligned} f &= (0\text{-}4\text{-}8\text{-}12) + (3\text{-}7\text{-}11\text{-}15) + (0\text{-}1) + (12\text{-}13) \\ f &= (0\text{-}4\text{-}8\text{-}12) + (3\text{-}7\text{-}11\text{-}15) + (0\text{-}1) + (13\text{-}15) \\ f &= (0\text{-}4\text{-}8\text{-}12) + (3\text{-}7\text{-}11\text{-}15) + (1\text{-}3) + (12\text{-}13) \\ f &= (0\text{-}4\text{-}8\text{-}12) + (3\text{-}7\text{-}11\text{-}15) + (1\text{-}3) + (13\text{-}15) \end{aligned}$$

Obtener las expresiones algebraicas de f resulta muy sencillo partiendo de las expresiones numéricas y recordando lo explicado en el apartado 3.2.

En efecto, la expresión algebraica de 0-4-8-12 se obtiene teniendo en cuenta que las variables que desaparecen en la expresión $\bar{a}\bar{b}\bar{c}\bar{d}$ del término canónico 0 son la de peso 4 ($4 - 0 = 4$), que es la c, y la de peso 8 ($8 - 0 = 8$), que es la d. Por tanto

$$0\text{-}4\text{-}8\text{-}12 \equiv \bar{a}\bar{b}$$

Las expresiones algebraicas de los restantes términos son:

$$\begin{aligned} 3\text{-}7\text{-}11\text{-}15 &\equiv a\,b \\ 0\text{-}1 &\equiv \bar{b}\,\bar{c}\,\bar{d} \\ 1\text{-}3 &\equiv a\,\bar{c}\,\bar{d} \\ 12\text{-}13 &\equiv \bar{b}\,c\,d \\ 13\text{-}15 &\equiv a\,c\,d \end{aligned}$$

y por tanto las cuatro expresiones algebraicas mínimas de la función f son:

$$f = \bar{a}\,\bar{b} + a\,b + \bar{b}\,\bar{c}\,\bar{d} + \bar{b}\,c\,d$$
$$f = \bar{a}\,\bar{b} + a\,b + \bar{b}\,\bar{c}\,\bar{d} + a\,c\,d$$
$$f = \bar{a}\,\bar{b} + a\,b + a\,\bar{c}\,\bar{d} + \bar{b}\,c\,d$$
$$f = \bar{a}\,\bar{b} + a\,b + a\,\bar{c}\,\bar{d} + a\,c\,d$$

Cuando la tabla de términos primos no esenciales contiene un número elevado de aquéllos es conveniente observar si alguna fila está cubierta por otra. Si todos los términos canónicos realizados por una fila son a su vez realizados por otra a la cual le corresponde un término primo de igual o menor complicación que aquélla, se dice que la primera está cubierta por la segunda y puede ser suprimida porque no formará parte de la solución mínima de la función.

En las tablas 3.5 y 3.6 se indica un ejemplo que aclara lo dicho.

La tabla 3.5 representa una tabla de términos primos no esenciales obtenida partiendo de la tabla general de términos primos, suprimiendo las filas correspondientes a los términos esenciales y las columnas cubiertas por ellos. Observando dicha tabla se comprueba que la línea *A* está cubierta por la *D* porque ésta realiza todos los términos de aquélla y a ambas le corresponden términos primos de igual complejidad. Igualmente la línea *B* cubre a la *C*. Por tanto pueden suprimirse las líneas *A* y *C* y la tabla resultante se indica en la tabla 3.6.

Términos primos no esenciales		2	3	8	10
A	0-1-2-3	X	X		
B	0-2-8-10	X		X	X
C	0-4-8-12			X	
D	2-3-10-11	X	X		X

TABLA 3.5

Términos primos no esenciales		2	3	8	10
B	0-2-8-10	X		X	X
D	2-3-10-11	X	X		X

TABLA 3.6

3.3 FUNCIONES INCOMPLETAS: DEFINICION Y APLICACION DE LOS METODOS DE SIMPLIFICACION

Hasta ahora se han estudiado funciones lógicas en las que para cada combinación de las entradas se define un valor lógico uno o cero de la función. Estas funciones se denominan totalmente definidas.

También existen funciones no totalmente definidas llamadas funciones incompletas, que son aquellas en las que, para una o más combinaciones de entrada, a la salida se le puede asignar el valor cero o el uno indistintamente. Esta situación puede presentarse por las dos causas siguientes:

1. No pueden existir una o más combinaciones de entrada. Por tanto, a la salida correspondiente a esas combinaciones se le puede asignar el valor cero o el valor uno.
2. Cuando aparecen una o más combinaciones de entrada, la acción de la salida del sistema lógico está inhibida. En consecuencia el valor de la salida para esas combinaciones de entrada también es indiferente.

La forma canónica de una función incompleta se representa indicando separadamente los términos canónicos para los cuales la función vale uno y los términos canónicos para los cuales es indiferente (lo cual se indica mediante el símbolo ϕ).

Sea, por ejemplo, la función incompleta

$$f(a, b, c, d) = \sum_{4} (1, 3, 6, 8, 10, 11) + \sum_{\phi} (0, 2, 4, 12, 13)$$

En la tabla de verdad de esta función que se representa en la tabla 3.7 se coloca un signo X en las posiciones de f correspondientes a las combinaciones de entrada para las cuales no está definido el estado de la salida.

La obtención de una de las expresiones canónicas de una función incompleta partiendo de la otra se realiza sin gran dificultad. La función inversa $\bar{f}$ será indiferente para las mismas combinaciones de entrada. Su expresión canónica en forma de suma de productos estará formada por una parte de términos indiferentes que coincide con la de f, y además serán uno los términos para los cuales f es cero.

Aplicando lo dicho al ejemplo de la tabla 3.7 resulta:

$$\bar{f} = \sum_{4} (5, 7, 9, 14, 15) + \sum_{\phi} (0, 2, 4, 12, 13)$$

Invirtiendo f y aplicando lo estudiado en el capítulo 2 resulta

$$\bar{\bar{f}} = f = \prod_{4} (0, 1, 6, 8, 10) \prod_{\phi} (2, 3, 11, 13, 15)$$

La minimización de estas funciones se puede realizar mediante cualquiera de los dos métodos que acabamos de estudiar.

La única diferencia con respecto a las funciones totalmente definidas es que a los términos indiferentes se les puede asignar a voluntad el valor cero o el valor uno. Por tanto el método tabular de Karnaugh se aplica de la misma forma que en el apartado 3.3 y, para realizar cada término de la función que toma el valor uno, se agrupan con él el máximo número de términos posible. Para lograrlo se asigna el valor uno a aquellos términos indiferentes que permitan simplificar la función y el valor cero a los restantes.

En la figura 3.11 se representan las tablas de Karnaugh de la expresión canónica en forma de suma de productos canónicos de la función cuya tabla de verdad se

d	c	b	a	f
0	0	0	0	X
0	0	0	1	1
0	0	1	0	X
0	0	1	1	1
0	1	0	0	X
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	X
1	1	0	1	X
1	1	1	0	0
1	1	1	1	0

TABLA 3.7

representa en la tabla 3.7. Se observa que existen dos expresiones mínimas de suma de productos representadas en las tablas *a* y *b* de la figura 3.11. La primera se obtiene realizando el producto canónico 8 mediante el término 0-2-8-10, y la segunda, realizándolo mediante el término 0-4-8-12. En la tabla *a* se asigna el valor cero a los productos canónicos indiferentes 12 y 13 y el valor uno a los productos canónicos 0, 2 y 4. En la tabla *b* se asigna el valor cero al producto canónico 13 y el valor uno a los productos canónicos 0, 2, 4 y 12.

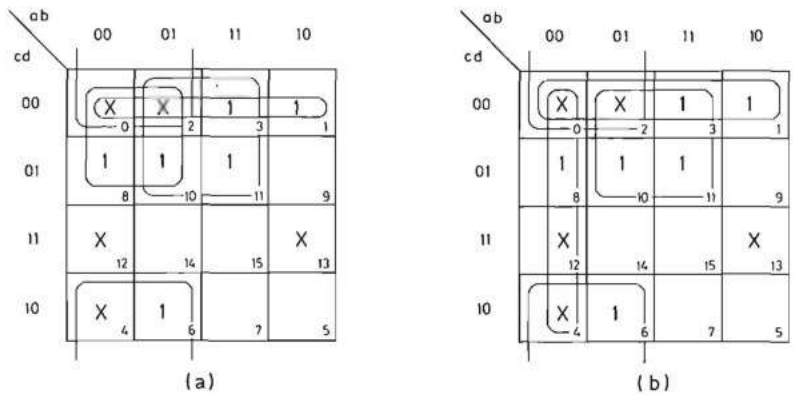


FIGURA 3.11. —**Tablas de Karnaugh de la función** $f = \sum_{4} (1, 3, 6, 8, 10, 11) + \sum_{\delta} (0, 2, 4, 12, 13)$.

Las expresiones algebraicas correspondientes a ambas tablas de la figura 3.11*a* y *b* son respectivamente:

$$f = \bar{a}\bar{c} + b\bar{c} + \bar{c}\bar{d} + \bar{a}\bar{d}$$
$$f = \bar{a}\bar{b} + b\bar{c} + \bar{c}\bar{d} + \bar{a}\bar{d}$$

La minimización de esta misma función por el método numérico se representa en las tablas 3.8 y 3.9. En la tabla 3.8 se representan las tablas de agrupación de los productos canónicos obtenidos de la forma explicada en el apartado 3.2.2. En la tabla 1 se incluyen los productos canónicos que toman el valor uno y los productos canónicos indiferentes y con ellos se obtienen los términos primos.

Tabla 1			Tabla 2			Tabla 3		Tabla 3 *reducida*	
N.º de unos					Dif.		Dif.		Dif.
0	0	X	0 - 1	X	1	0 - 1 - 2 - 3	1,2	0 - 1 - 2 - 3	1,2
	1	X	0 - 2	X	2	0 - 2 - 1 - 3	2,1	0 - 2 - 4 - 6	2,4
	2	X	0 - 4	X	4	0 - 2 - 4 - 6	2,4	0 - 2 - 8 -10	2,8
1	4	X	0 - 8	X	8	0 - 2 - 8 -10	2,8	0 - 4 - 8 -12	4,8
	8	X	1 - 3	X	2	0 - 4 - 2 - 6	4,2	2 - 3 -10 -11	1,8
	3	X	2 - 3	X	1	0 - 4 - 8 -12	4,8		
	6	X	2 - 6	X	4	0 - 8 - 2 -10	8,2		
2	10	X	2 -10	X	8	0 - 8 - 4 -12	8,4		
	12	X	4 - 6	X	2	2 - 3 -10 -11	1,8		
	11	X	4 -12	X	8	2 -10 - 3 -11	8,1		
3	13	X	8 -10	X	2				
			8 -12	X	4				
			3 -11	X	8				
			10 -11	X	1				
			12 -13		1				

TABLA 3.8

Términos primos	1	3	6	8	10	11
0 - 1 - 2 - 3	⊗	X				
0 - 2 - 4 - 6			⊗			
0 - 2 - 8 -10				X	X	
0 - 4 - 8 -12				X		
2 - 3 -10 -11		X			X	⊗

TABLA 3.9

Las columnas de la tabla de términos primos de la tabla 3.9 corresponden a los productos canónicos que es necesario realizar y que son únicamente los que toman el valor uno. De acuerdo con lo explicado en el apartado 3.2.2, se observa que los términos 0-1-2-3, 0-2-4-6 y 2-3-10-11 son esenciales. Sin necesidad de hacer otra tabla o utilizar la ecuación algebraica correspondiente se observa que el único producto canónico que queda por realizar es el 8 y que para ello se pueden utilizar los términos 0-2-8-10 o 0-4-8-12. Por tanto existen las soluciones mínimas.

$$f = (0\text{-}1\text{-}2\text{-}3) + (0\text{-}2\text{-}4\text{-}6) + (2\text{-}3\text{-}10\text{-}11) + (0\text{-}2\text{-}8\text{-}10)$$
$$f = (0\text{-}1\text{-}2\text{-}3) + (0\text{-}2\text{-}4\text{-}6) + (2\text{-}3\text{-}10\text{-}11) + (0\text{-}4\text{-}8\text{-}12)$$

Observando las tablas de la figura 3.11 se comprueba que estas dos soluciones coinciden con las obtenidas mediante el método tabular de Karnaugh.

Los ejemplos se han realizado utilizando la suma de productos canónicos, pero los métodos que se han empleado son también aplicables al producto de sumas canónicas.

En la tabla 3.7 se representa la tabla de verdad de una función incompleta y en ella se indican todas las combinaciones posibles de las variables de entrada. Cuando una función f no está definida para un número elevado de las combinaciones de entrada no hay que representar la tabla de verdad completa. No es necesario indicar las combinaciones de entrada para las que la función no está definida. Y además cuando la función f toma un cierto valor para una combinación de determinadas variables de entrada independientemente del estado de las demás, esta situación se puede representar mediante una sola línea en la tabla de verdad.

La tabla 3.10 es un ejemplo de lo que se acaba de exponer. La función f adopta el estado cero cuando $a = b = c = d = 0$ independientemente del estado de e y g. Esto se indica mediante la primera línea de la tabla 3.10 en la que se coloca el símbolo de indiferencia X en las columnas de las variables e y g. En esta tabla no se indican todos los términos correspondientes a las combinaciones de las variables

g	e	d	c	b	a	f
X	X	0	0	0	0	0
X	0	0	0	0	1	1
X	1	0	0	1	0	0
X	X	0	0	1	1	0
X	X	0	1	0	0	1
X	X	0	1	0	1	1
X	1	0	1	1	0	1
1	X	0	1	1	1	0
0	X	1	0	0	0	1
X	X	1	0	0	1	0

TABLA 3.10

a, b, c y d superiores a 1001 porque se supone que son indiferentes. De esta forma se reduce considerablemente el tamaño de la tabla de verdad.

El método simplificado de Karnaugh descrito en el apartado 3.2.1 es aplicable también a las funciones incompletas. Para ello se deben asignar a las filas y columnas de la tabla de Karnaugh aquellas variables que están definidas en la mayoría de las combinaciones de la tabla. Por ejemplo, si se trata de obtener la expresión más sencilla de la función cuya tabla de verdad se representa en la tabla 3.10 mediante una tabla de Karnaugh de 4 variables, se deben asignar a las filas y columnas las variables a, b, c y d que son las que están definidas en todas las líneas de la tabla.

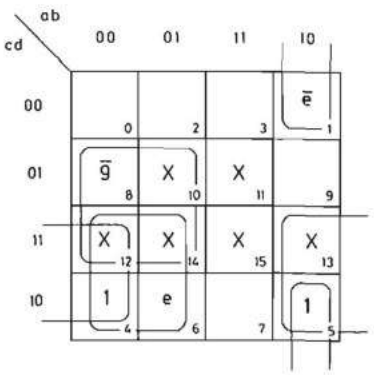


FIGURA 3.12.—**Tabla de Karnaugh única de la función representada en la tabla 3.10.**

En la figura 3.12 se representa la tabla de Karnaugh de suma de productos de cuatro variables. A cada cuadrado se le asignan los números 0 a 15 correspondientes a los dieciséis primeros productos canónicos de la función f. En los cuadrados 10 a 15 se coloca el símbolo X porque la función f no está definida para las combinaciones de entrada correspondientes. En los cuadrados 4 y 5 se coloca un uno porque la función f adopta el estado uno para las combinaciones 0100 y 0101 de las variables d, c, b y a independientemente del estado de e y g. En el cuadrado 1 se coloca la variable $\bar{e}$ porque para la combinación 0001 de las variables d, c, b y a la función adopta el estado uno cuando $e = 0$. De igual forma en el cuadrado 8 se coloca la variable $\bar{g}$ y en el cuadrado 6 la variable e.

A continuación se realizan los grupos necesarios para cubrir todos los unos de la manera más sencilla posible. Se obtiene así el grupo $4 - 5 - 12 - 13$ que corresponde al producto $\bar{b}c$. Seguidamente se realizan las variables $\bar{e}, \bar{g}$ y e combinándolas con las X y los unos.

Se obtienen así los grupos:

$$\begin{aligned} 8 - 10 - 12 - 14 &= \bar{a}d\bar{g} \\ 4 - 6 - 12 - 14 &= \bar{a}ce \\ 1 - 5 &= a\bar{b}\bar{d}\bar{e} \end{aligned}$$

Por lo tanto la expresión mínima de sumas de productos de la función f resulta:

$$f = \bar{b}c + \bar{a}d\bar{g} + \bar{a}ce + a\,\bar{b}\,\bar{d}\,\bar{e}$$

3.4 MULTIFUNCIONES: DEFINICION Y APLICACION DE LOS METODOS DE SIMPLIFICACION

En apartados anteriores se ha estudiado la simplificación de los circuitos combinacionales equivalentes a una sola función lógica y realizado diversos ejemplos. Con frecuencia, los circuitos combinacionales poseen más de una salida, es decir están constituidos por varias funciones lógicas que dependen de las mismas variables de entrada y han de ser realizadas simultáneamente. Estos circuitos combinacionales son por tanto equivalentes a una multifunción.

La minimización de una multifunción se puede realizar tratando cada una de las funciones independientemente por los métodos indicados en los apartados anteriores, pero con ello no se tiene la seguridad de obtener el circuito más sencillo y por lo tanto más económico. Puede ser conveniente complicar la expresión lógica de las funciones independientes a fin de que tengan productos o sumas comunes entre sí y de que el conjunto implique un menor número de elementos en su realización.

Las multifunciones pueden ser totalmente definidas o no totalmente definidas. El tratamiento de ambos casos se diferencia en lo que ya se ha indicado en el apartado 3.3 para las funciones sencillas y, por tanto, no es preciso volver a analizarlo.

También pueden simplificarse las multifunciones por el método tabular de Karnaugh o el método numérico. Tanto en uno como en otro es necesario tener en cuenta los términos comunes a más de una función.

Por el método de Karnaugh se han de realizar las siguientes etapas en la minimización:

a) Realización de tablas de todas las funciones y sus productos lógicos.

b) Todos aquellos términos canónicos que sólo toman el valor uno para una función se realizan de la forma más sencilla posible en la tabla correspondiente a dicha función.

c) Todos aquellos términos que son comunes a dos funciones, es decir que forman parte del producto lógico de ambas, que no son comunes a tres funciones y que no han sido realizados en el apartado anterior, se realizan de la forma más sencilla posible en la tabla correspondiente.

d) El proceso se repite para los términos comunes a tres funciones que todavía no han sido realizados y así sucesivamente.

De lo dicho se deduce que el método tabular de Karnaugh no resulta aplicable en la práctica a multifunciones que incluyan más de tres funciones simples.

Un ejemplo aclarará todo lo expuesto. En la figura 3.13 se representan las tablas de Karnaugh de la multifunción cuya tabla de verdad se representa en la tabla 3.11.

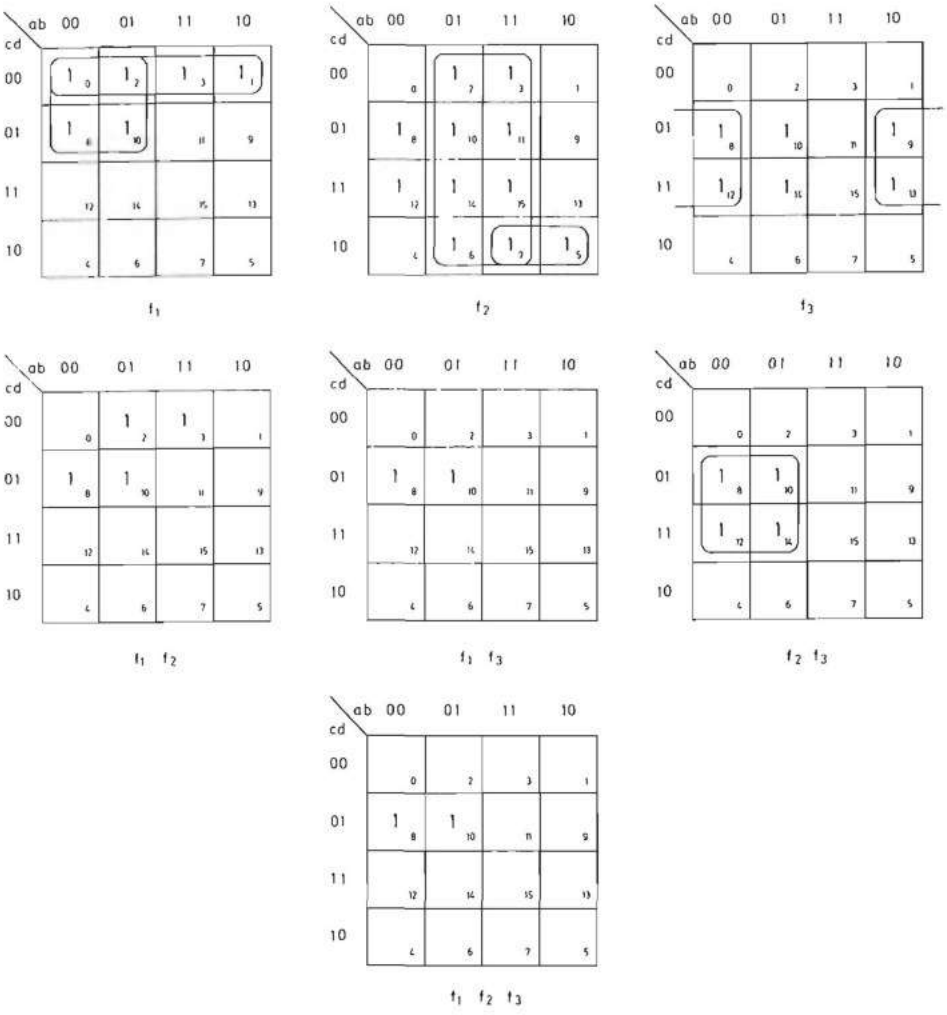


FIGURA 3.13.—**Tablas de Karnaugh de las funciones de la tabla 3.11.**

Tabla de verdad de la multifunción:

$$f_1(abcd) = \sum_4 (0, 1, 2, 3, 8, 10)$$

$$f_2(abcd) = \sum_4 (2, 3, 5, 6, 7, 8, 10, 11, 12, 14, 15)$$

$$f_3(abcd) = \sum_4 (8, 9, 10, 12, 13, 14)$$

Analizando la tabla de cada función se observa, por ejemplo, que los productos canónicos 0 y 1 sólo toman parte de la función f_1 y se realizan de la forma más sencilla posible mediante el término 0-1-2-3. Igualmente los productos canónicos

d	c	b	a	f_1	f_2	f_3
0	0	0	0	1	0	0
0	0	0	1	1	0	0
0	0	1	0	1	1	0
0	0	1	1	1	1	0
0	1	0	0	0	0	0
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	1	1
1	0	0	1	0	0	1
1	0	1	0	1	1	1
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	0	1	0	0	1
1	1	1	0	0	1	1
1	1	1	1	0	1	0

TABLA 3.11

6, 7, 11 y 15 sólo forman parte, al igual que el 5, de la función f_2 y se realizan respectivamente mediante los términos 2-3-6-7-10-11-14-15 y 5-7. De igual manera, los términos 9 y 13 sólo pertenecen a f_3 y se realizan mediante el término 8-9-12-13.

Una vez realizado lo anterior se observan las tablas correspondientes al producto de dos funciones. Los productos canónicos 12 y 14 que todavía no han sido realizados y pertenecen a $f_2 f_3$ no están en $f_1 f_2 f_3$ y por tanto se realizan mediante el término 8-10-12-14.

Sólo quedan por realizar, por tanto, los productos canónicos 8 y 10 de f_1 para lo cual se utilizará el producto 8-10 que está en $f_1 f_2 f_3$. Pero como este término sólo formará parte de f_1 porque ya ha sido realizado en f_2 y f_3 mediante 8-10-12-14, se puede simplificar uniéndolo al 0-2 para formar el término 0-2-8-10.

Por tanto, las funciones mínimas resultantes son:

$$f_1 = \bar{c}\bar{d} + \bar{a}\bar{c}$$
$$f_2 = b + ac\bar{d} + \bar{a}d$$
$$f_3 = \bar{b}d + \bar{a}d$$

La minimización de las multifunciones por el método numérico presenta la enorme ventaja de ser aplicable en cualquier caso independientemente de su complejidad, en especial mediante un programa de computador.

En la tabla de términos (productos o sumas) canónicos ordenados por el número de unos se indica, a la derecha del término correspondiente, a qué funciones pertenece. Mediante un ejemplo se aclaran mejor los conceptos. Se utiliza el mismo al que se le aplicó el método tabular de Karnaugh y cuya tabla de verdad

Tabla 1

N.º de unos			
0	0	X	f_1
	1	X	f_1
1	2	X	f_1f_2
	8	X	$f_1f_2f_3$
	3	X	f_1f_2
	5	X	f_2
2	6	X	f_2
	9	X	f_3
	10	X	$f_1f_2f_3$
	12	X	f_2f_3
	7	X	f_2
	11	X	f_2
3	13	X	f_3
	14	X	f_2f_3
4	15	X	f_2

Tabla 2

			Dif.
0-1	X	f_1	1
0-2	X	f_1	2
0-8	X	f_1	8
1-3	X	f_1	2
~~1-5~~		-	
~~1-9~~		-	
2-3		f_1f_2	1
2-6	X	f_2	4
2-10		f_1f_2	8
8-9	X	f_3	1
8-10		$f_1f_2f_3$	2
8-12	X	f_2f_3	4
3-7	X	f_2	4
3-11	X	f_2	8
5-7		f_2	2
~~5-13~~		-	
6-7	X	f_2	1
6-14	X	f_2	8
~~9-11~~		-	
9-13	X	f_3	4
10-11	X	f_2	1
10-14	X	f_2f_3	4
12-13	X	f_3	1
12-14	X	f_2f_3	2
7-15	X	f_2	8
11-15	X	f_2	4
~~13-15~~		-	
14-15	X	f_2	1

Tabla 3

		Dif.
0-1-2-3	f_1	1,2
~~0-1-8-9~~	-	
0-2-1-3	f_1	2,1
0-2-8-10	f_1	2,8
0-8-2-10	f_1	8,2
~~1-3-5-7~~	-	
2-3-6-7	f_2	1,4
2-3-10-11	f_2	1,8
2-6-3-7	f_2	4,1
2-6-10-14	f_2	4,8
2-10-3-11	f_2	8,1
2-10-6-14	f_2	8,4
~~8-9-10-11~~	-	
8-9-12-13	f_3	1,4
8-10-12-14	f_2f_3	2,4
8-12-10-14	f_2f_3	4,2
8-12-9-13	f_3	4,1
3-7-11-15	f_2	4,8
3-11-7-15	f_2	8,4
6-7-14-15	f_2	1,8
6-14-7-15	f_2	8,1
~~9-13-11-15~~	-	
10-11-14-15	f_2	1,4
10-14-11-15	f_2	4,1
~~12-13-14-15~~	-	

Tabla 3 *reducida*

			Dif.
0-1-2-3		f_1	1,2
0-2-8-10		f_1	2,8
2-3-6-7	X	f_2	1,4
2-3-10-11	X	f_2	1,8
2-6-10-14	X	f_2	4,8
8-9-12-13		f_3	1,4
8-10-12-14		f_2f_3	2,4
3-7-11-15	X	f_2	4,8
6-7-14-15	X	f_2	1,8
10-11-14-15	X	f_2	1,4

Tabla 4

		Dif.
2-3-6-7-10-11-14-15	f_2	1,4,8
2-3-10-11-6-7-14-15	f_2	1,8,4
2-6-10-14-3-7-11-15	f_2	4,8,1
~~8-9-12-13-10-11-14-15~~	-	

Tabla 4 *reducida*

		Dif.
2-3-6-7-10-11-14-15	f_2	1,4,8

TABLA 3.12

se representa en la tabla 3.11. En la tabla 3.12 se indican todas las tablas obtenidas. Las tablas 2 y 3 se realizan por el método ya explicado en el apartado 3.2.2 indicando las funciones a las que pertenece cada término. Por ejemplo, el término 12-13 pertenece a f_3 porque el 12 pertenece a f_2 y f_3 y el 13 sólo a f_3. El término 1-5, por

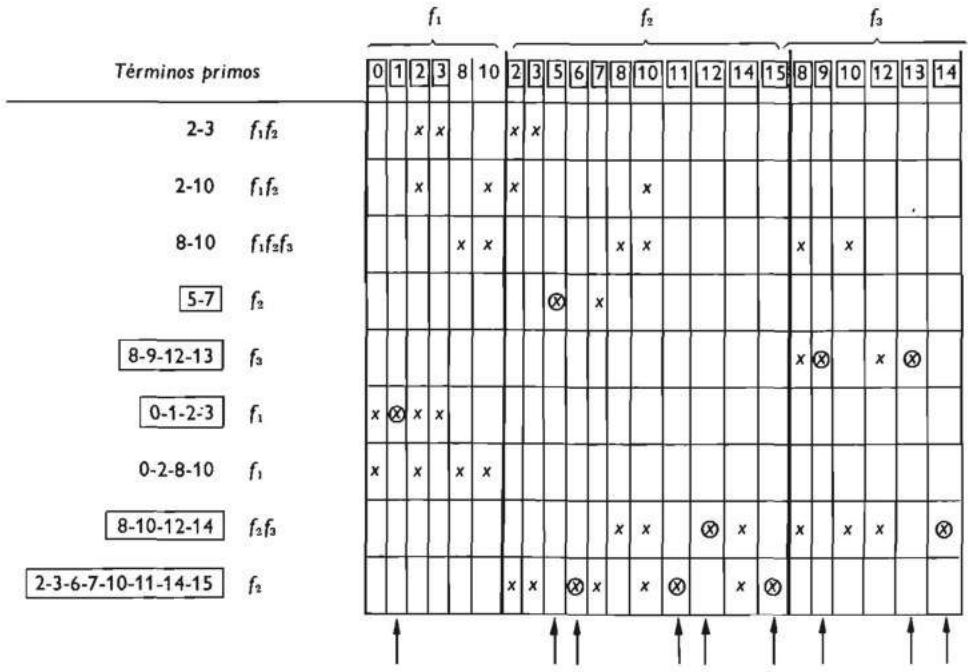


TABLA 3.13.—**Tabla de términos primos.**

ejemplo, se suprime porque el 1 pertenece a f_1 y el 5 a f_2. Los términos suprimidos se cubren con un aspa. Para que un término de cualquiera de ellas sea primo y por tanto utilizable en la realización de la función, es necesario que no exista ninguno que lo contenga y que pertenezca a las mismas funciones que él. Por ejemplo, el término 8-10 es primo porque pertenece a f_1, f_2 y f_3, mientras el 0-2-8-10 sólo pertenece a f_1 y el 8-10-12-14 sólo a f_2 y f_3.

La tabla de los términos primos se presenta en la tabla 3.13 cuyas columnas corresponden a todos los términos canónicos de las tres funciones. En esta tabla se observa que los términos 1 de f_1, 5, 6, 11, 12, 15 de f_2 y 9, 13, 14 de f_3 son realizados por un solo término primo, el cual es pues esencial. El lector puede comprobar que los términos primos esenciales son 5-7(f_2), 8-9-12-13(f_3) 0-1-2-3(f_1), 8-10-12-14(f_2f_3) y 2-3-6-7-10-11-14-15(f_2) y que los únicos términos canónicos no realizados por ellos son el 8 y el 10 de f_1. Estos dos términos canónicos están cubiertos por los términos primos 2-10, 8-10 y 0-2-8-10, obteniéndose la tabla 3.14 en la cual se observa que la línea *C* cubre a las líneas *A* y *B*. Por tanto, a los términos esenciales hay que añadir el 0-2-8-10 (f_1).

Las expresiones resultantes de las tres funciones son:

$$f_1 = (0\text{-}1\text{-}2\text{-}3) + (0\text{-}2\text{-}8\text{-}10)$$
$$f_2 = (5\text{-}7) + (8\text{-}10\text{-}12\text{-}14) + (2\text{-}3\text{-}6\text{-}7\text{-}10\text{-}11\text{-}14\text{-}15)$$
$$f_3 = (8\text{-}9\text{-}12\text{-}13) + (8\text{-}10\text{-}12\text{-}14)$$

	Términos primos no esenciales		f_1	
			8	10
A	2 - 10	$f_1 f_2$	X	
B	8 - 10	$f_1 f_2 f_3$	X	X
C	0 - 2 - 8 - 10	f_1	X	X

TABLA 3.14

cuyo equivalente algebraico es:

$$f_1 = \bar{c}\bar{d} + \bar{a}\bar{c}$$
$$f_2 = b + ac\bar{d} + \bar{a}d$$
$$f_3 = \bar{b}d + \bar{a}d$$

Se observa que estas expresiones coinciden con las obtenidas mediante el método tabular de Karnaugh.

3.5 REALIZACION DE LAS FUNCIONES LOGICAS

3.5.1 Realización con puertas NO-Y (NAND) y NO-O (NOR)

Una vez que se ha obtenido la expresión mínima de una función es necesario realizarla en la práctica mediante elementos físicos. El diseño de puertas lógicas con transistores en un principio y la posterior aparición de los circuitos integrados ha hecho que las puertas NO-Y y NO-O sean las más utilizadas en la realización de las funciones lógicas. En el capítulo 2 se ha demostrado que las funciones NO-Y y NO-O pueden realizar cualquiera de las tres funciones elementales suma, producto, e inversión.

Para realizar con puertas NO-Y (NO-O) la expresión mínima de la función obtenida por el método tabular o el método numérico, se aplicarán las siguientes reglas cuya validez se deduce de los postulados y teoremas estudiados en el capítulo 2.

a) Se aplican a la expresión global de la función dos inversiones con lo cual la misma queda invariable.

b) Si la operación más externa es una suma (producto) lógica, se opera una de las inversiones aplicando el teorema de De Morgan y si es producto (suma) no se operan ninguna de las dos.

c) Si en el interior de la expresión existen sumas (productos) lógicas, se aplican a cada una de ellas dos inversiones y se opera una de ellas para convertirla en el inverso de un producto (suma).

d) Se continúa realizando esta operación hasta que todas las sumas (productos) hayan quedado convertidas en inversos de productos (sumas).

Las reglas para realizar cualquier expresión con puertas NO-O (NOR) son iguales a las de la puerta NO-Y (NAND) sustituyendo la palabra suma por producto, lo cual se ha indicado incluyendo la palabra suma entre paréntesis en las reglas que acabamos de indicar.

Unos cuantos ejemplos aclararán estas reglas. Se utilizarán para ello las expresiones de las funciones simplificadas en apartados anteriores.

Las expresiones obtenidas en el ejemplo 3.2 fueron:

$$f_1 = b\bar{c} + ab + ac\bar{d}$$
$$f_1 = (a + \bar{c})\,(b + c)\,(b + \bar{d})$$

Se realizan primero con NO-Y y después con NO-O aplicando las reglas anteriores:

$$f_1 = b\bar{c} + ab + ac\bar{d} = \overline{\overline{b\bar{c} + ab + ac\bar{d}}} = \overline{\overline{b\bar{c}}\ \ \overline{ab}\ \ \overline{ac\bar{d}}}$$

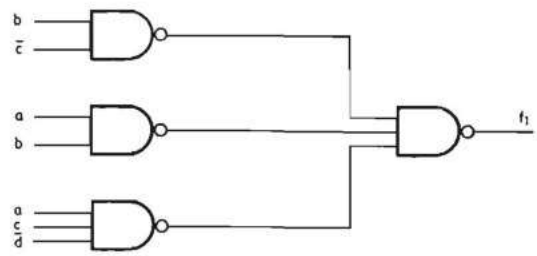


FIGURA 3.14. —**Esquema del circuito con puertas NO-Y (NAND) de la función $f_1 = b\,\bar{c} + a\,b + a\,c\,\bar{d}$, realizado con símbolos no normalizados.**

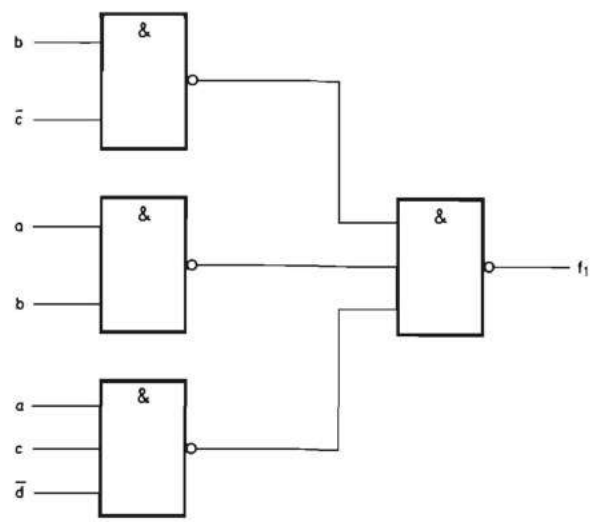


FIGURA 3.15. —**Esquema del circuito con puertas NO-Y (NAND) de la función $f_1 = b\,c + a\,b + a\,c\,\bar{d}$, realizado con símbolos normalizados.**

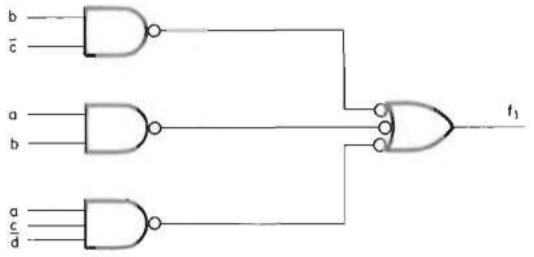


FIGURA 3.16.—**Esquema equivalente al de la figura 3.14 realizado con símbolos no normalizados.**

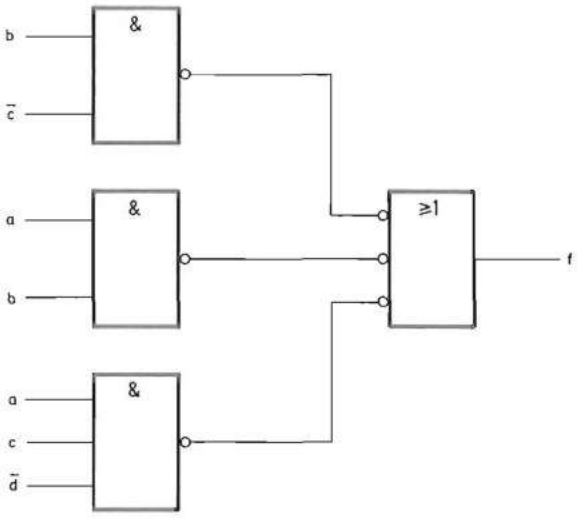


FIGURA 3.17.—**Esquema equivalente al de la figura 3.15 realizado con símbolos normalizados.**

En la figura 3.14 se representa el esquema lógico correspondiente con los símbolos utilizados hasta ahora y en la figura 3.15 con los nuevos normalizados. Antes de continuar el estudio de este capítulo se recomienda al lector que analice los apartados A1.1, A1.2, A1.3, A1.4.1 y A1.4.2 del apéndice 1 dedicado a los símbolos lógicos normalizados.

Se supone en este ejemplo y en lo sucesivo que se dispone tanto de las variables inversas como de las directas.

En el apartado 2.2 se indican los dos símbolos posibles de la función NO-Y (NAND). La combinación de ambos ayuda a interpretar mejor los esquemas. Como regla general se puede decir que las puertas NO-Y mediante las que se realizan los productos lógicos se deben representar con el símbolo de la puerta Y seguida de una inversión. Las puertas NO-Y que realizan las sumas lógicas se deben representar por medio de la puerta O precedida de inversiones en sus entradas.

Se obtienen de esta forma los esquemas de las figuras 3.16 y 3.17 que son equivalentes a los de las figuras 3.14 y 3.15.

Para realizar la expresión de producto de sumas con puertas NO-Y (NAND) es necesario también transformarla:

$$f_1 = (a + \bar{c})\,(b + c)\,(b + \bar{d}) = \overline{\overline{(a + \bar{c})\,(b + c)\,(b + \bar{d})}} = \overline{\overline{a + \bar{c}}\;\;\overline{b + c}\;\;\overline{b + \bar{d}}} =$$

$$= \overline{\bar{a}c\;\;\bar{b}\bar{c}\;\;\bar{b}d}$$

Esta expresión se representa en las figuras 3.18 y 3.19 y, se observa que son más complejas que la de las figuras 3.14 y 3.15. Combinando ambos tipos de símbolos se obtienen los esquemas de las figuras 3.20 y 3.21 que son equivalentes a los de las figuras 3.18 y 3.19.

De igual forma, ambas expresiones se realizan con puertas NO-O.

$$f_1 = b\bar{c} + ab + ac\bar{d} = \overline{\overline{b\bar{c} + ab + ac\bar{d}}} =$$

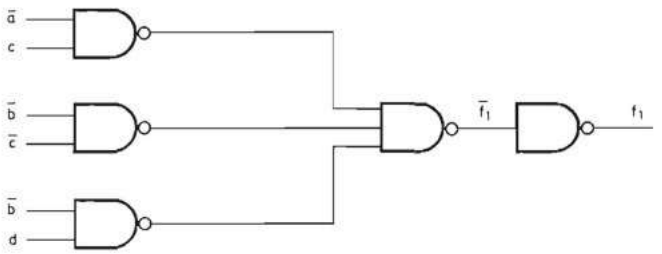


FIGURA 3.18. —**Esquema del circuito con puertas NO-Y (NAND) de la expresión de la función $f_1 = (\bar{a} + c)\,(\bar{b} + \bar{c})\,(\bar{b} + d)$, realizado con símbolos no normalizados.**

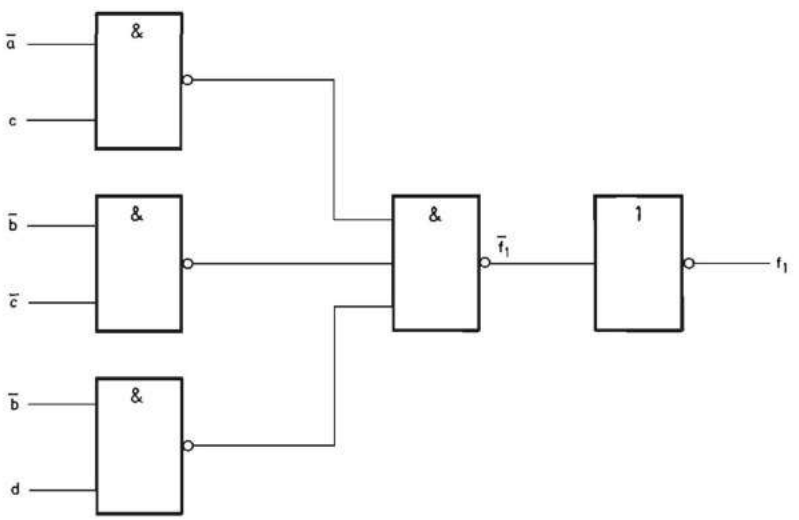


FIGURA 3.19. —**Esquema del circuito con puertas NO-Y (NAND) de la expresión de la función $f_1 = (\bar{a} + c)\,(\bar{b} + \bar{c})\,(\bar{b} + d)$, realizado con símbolos normalizados.**

$$= \overline{\overline{b\bar{c}} + \overline{ab} + \overline{acd}} = \overline{\overline{b} + c + \overline{\bar{a}} + \overline{b} + \overline{\bar{a}} + \overline{\bar{c}} + d}$$

$$f_1 = (a + \bar{c})\,(b + c)\,(b + \bar{d}) = \overline{\overline{(a + \bar{c})\,(b + c)\,(b + \bar{d})} =$$

$$= \overline{\overline{a + \bar{c}} + \overline{b + c} + \overline{b + \bar{d}}}$$

En las figuras 3.22 y 3.23 se representa la primera expresión y en las figuras 3.24 y 3.25 la segunda.

Al igual que la función NO-Y (NAND), la función NO-O (NOR) se puede representar gráficamente mediante dos símbolos diferentes indicados en las figuras 2.4 y 2.5, y la combinación de ambos también ayuda a interpretar mejor los esquemas.

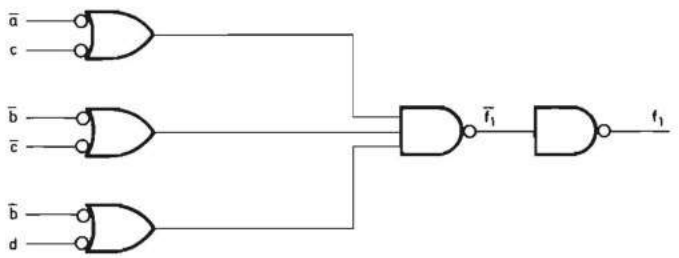


FIGURA 3.20.—**Esquema equivalente al de la figura 3.18 realizado con símbolos no normalizados.**

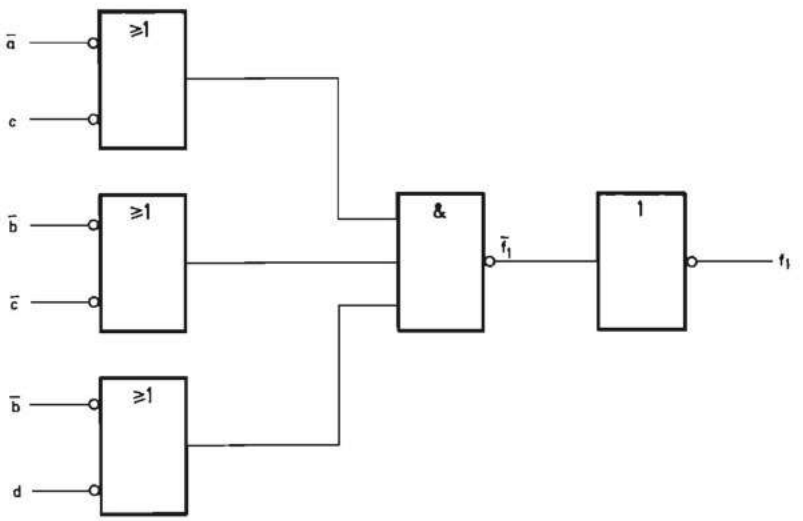


FIGURA 3.21.—**Esquema equivalente al de la figura 3.19 realizado con símbolos normalizados.**

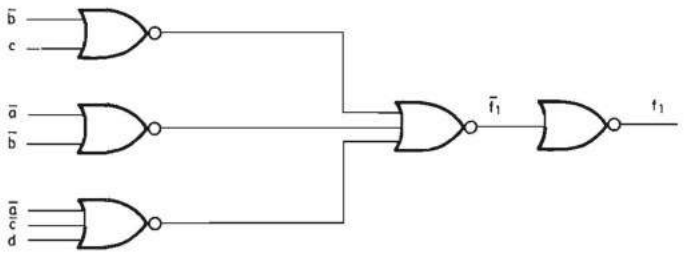


FIGURA 3.22. —**Esquema con puertas NO-O (NOR) de la expresión de la función** $f_1 = \overline{b}\,c + \overline{a}\,\overline{b} + \overline{a}\,\overline{c}\,d$, **realizado con símbolos no normalizados.**

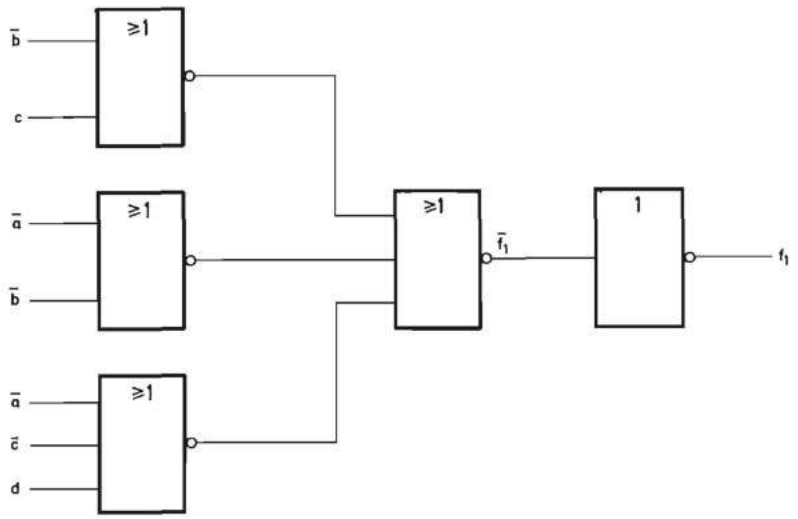


FIGURA 3.23. —**Esquema con puertas NO-O (NOR) de la expresión de la función** $f_1 = \overline{b}\,c + \overline{a}\,\overline{b} + \overline{a}\,\overline{c}\,d$, **realizado con símbolos normalizados.**

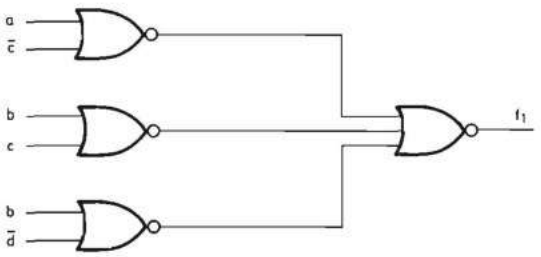


FIGURA 3.24. —**Esquema con puertas NO-O (NOR) de la expresión de la función** $f_1 = (a + \overline{c})\,(b + c)\,(b + \overline{d})$, **realizado con símbolos no normalizados.**

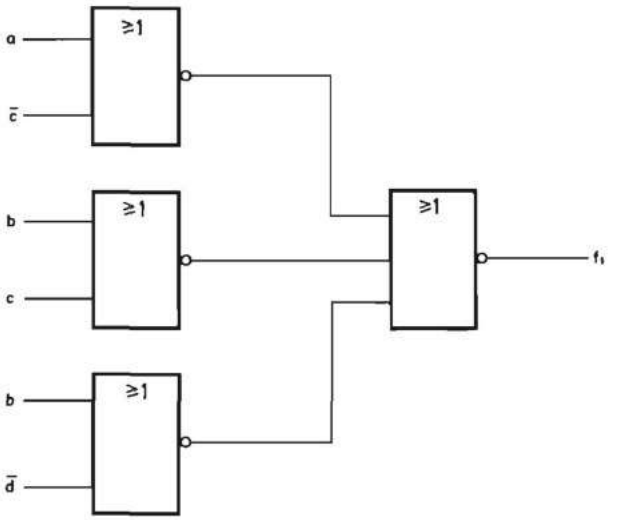


FIGURA 3.25.—**Esquema con puertas NO-O (NOR) de la expresión de la función $f_1 = (a + \bar{c})\ (b + c)\ (b + \bar{d})$, realizado con símbolos normalizados.**

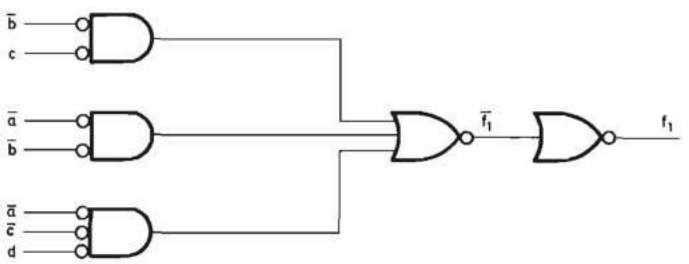


FIGURA 3.26.—**Esquema equivalente al de la figura 3.22 realizado con símbolos no normalizados.**

Es preferible representar con el símbolo de la puerta Y precedida de inversiones aquellas puertas que realizan los productos y mediante la puerta O seguida de una inversión las que realizan las sumas. Se obtienen de esta forma las figuras 3.26 y 3.27 equivalentes a las 3.22 y 3.23 respectivamente y las 3.28 y 3.29 equivalentes a las 3.24 y 3.25.

No teniendo en cuenta el inversor final cuando existe, en las figuras 3.14 a 3.29 se observa que para que el cambio de cualquier entrada actúe sobre una salida es necesario que conmuten como máximo dos puertas. Cada puerta representa un retraso a la conmutación y se denomina nivel de un circuito el máximo número de puertas que ha de atravesar la información del cambio de una cualquiera de las entradas para actuar sobre la salida. Se observa que, realizando las expresiones mínimas que se han obtenido por el método tabular o el numérico, se obtienen circuitos de nivel 2.

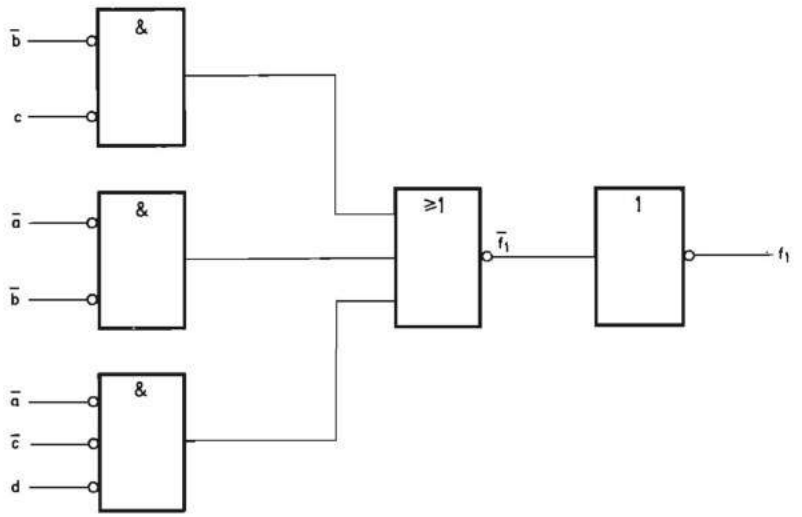


FIGURA 3.27.—**Esquema equivalente al de la figura 3.23 realizado con símbolos normalizados.**

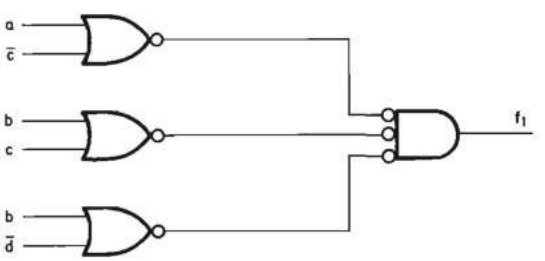


FIGURA 3.28.—**Esquema equivalente al de la figura 3.24 realizado con símbolos no normalizados.**

A dichas expresiones se les puede aplicar las propiedades distributivas de la suma con respecto al producto y viceversa. De esta forma se puede lograr una expresión más sencilla que se realizará con un menor número de puertas o el mismo número de puertas con menos entradas, pero que dará, en general, lugar a un circuito cuyo nivel será superior a 2. Naturalmente, existirá más de una forma de aplicar dichas propiedades, y por tanto, se obtendrá más de una expresión. Se utilizará el ejemplo analizado en párrafos anteriores para aclarar lo expuesto.

En la expresión de f_1 en forma de suma de productos:

$$f_1 = b\bar{c} + ab + ac\bar{d}$$

se observa que *b* es común a los dos primeros y *a* lo es a los dos segundos. Sacando factor común «*b*» resulta la expresión:

$$f_1 = b(a + \bar{c}) + ac\bar{d}$$

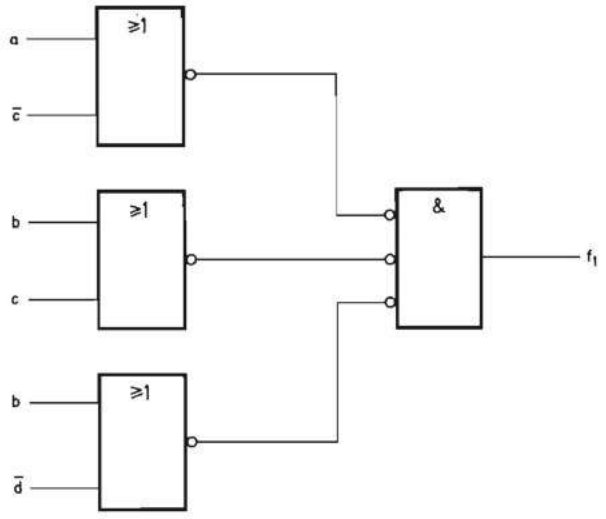


FIGURA 3.29.—**Esquema equivalente al de la figura 3.25 realizado con símbolos normalizados.**

y sacando factor común «a» resulta:

$$f_1 = b\bar{c} + a(b + c\bar{d})$$

Aplicando las reglas antes indicadas se pueden convertir estas expresiones para ser realizadas con puertas NO-Y y NO-O. Con puertas NO-Y resulta:

$$f_1 = b(a + \bar{c}) + ac\bar{d} = \overline{\overline{b(a + \bar{c}) + ac\bar{d}}} = \overline{\overline{b(a + \bar{c})}\ \overline{ac\bar{d}}} = \overline{b\ \overline{\overline{a + \bar{c}}}\ \overline{ac\bar{d}}} =$$

$$= \overline{\overline{b\bar{a}c}\ \overline{ac\bar{d}}}$$

$$f_1 = b\bar{c} + a(b + c\bar{d}) = \overline{\overline{b\bar{c} + a(b + c\bar{d})}} = \overline{\overline{b\bar{c}}\ \overline{a(b + c\bar{d})}} = \overline{\overline{b\bar{c}}\ a\ \overline{b + c\bar{d}}} =$$

$$= \overline{\overline{b\bar{c}}\ a\ \overline{\bar{b}\ \overline{c\bar{d}}}}$$

En lo sucesivo se utilizan exclusivamente en los esquemas los nuevos símbolos normalizados por la Comisión Electrotécnica Internacional.

En las figuras 3.30 y 3.31 se representan los esquemas correspondientes a las dos últimas expresiones de f_1.

De igual forma, representando las puertas NO-Y que realizan las sumas de las expresiones anteriores mediante el símbolo de la puerta O precedido de inversiones, se obtienen los esquemas de las figuras 3.32 y 3.33 equivalentes respectivamente a los de las figuras 3.30 y 3.31.

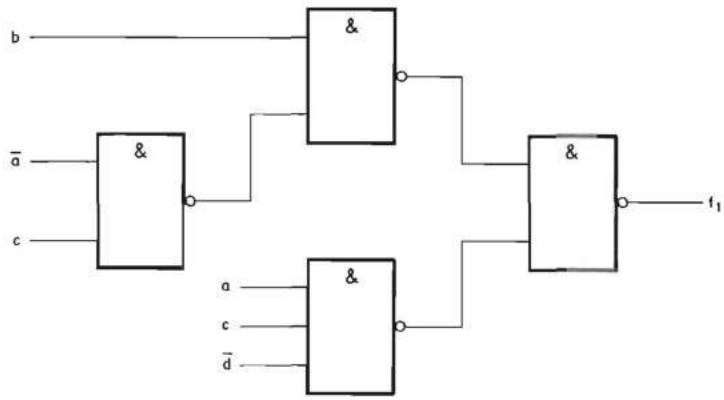


FIGURA 3.30. —**Esquema con puertas NO-Y (NAND) de la expresión de la función** $f_1 = b\ (a + \overline{c}) + a\ c\ \overline{d}$.

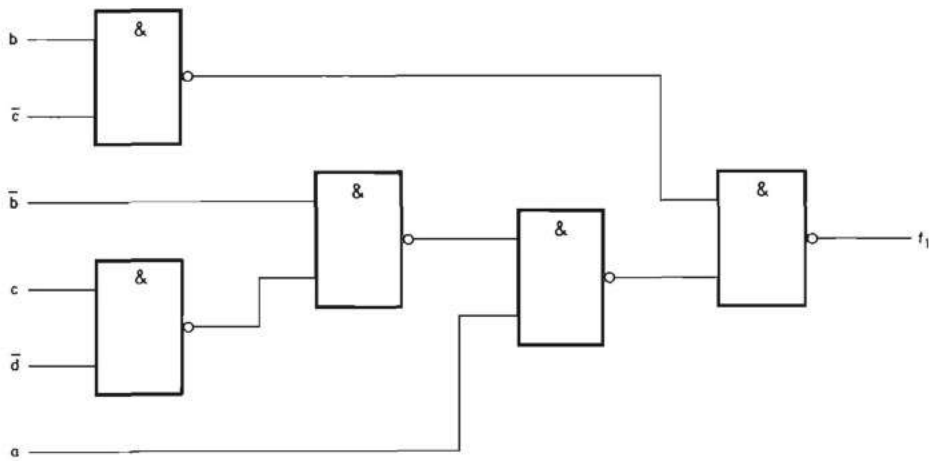


FIGURA 3.31. —**Esquema con puertas NO-Y (NAND) de la expresión de la función** $f_1 = b\ \overline{c} + a\ (b + c\ \overline{d})$.

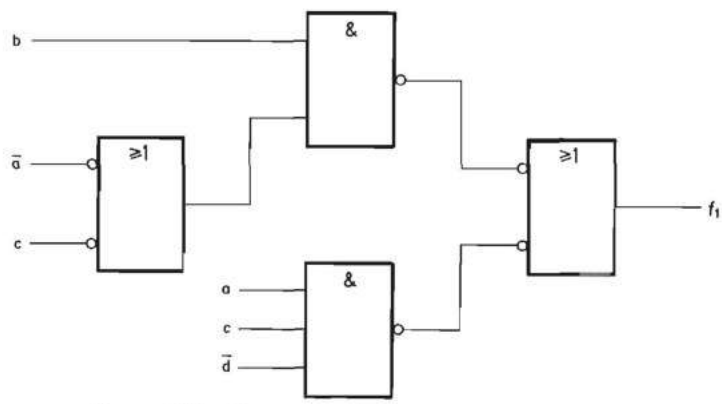


FIGURA 3.32.—**Esquema equivalente al de la figura 3.30.**

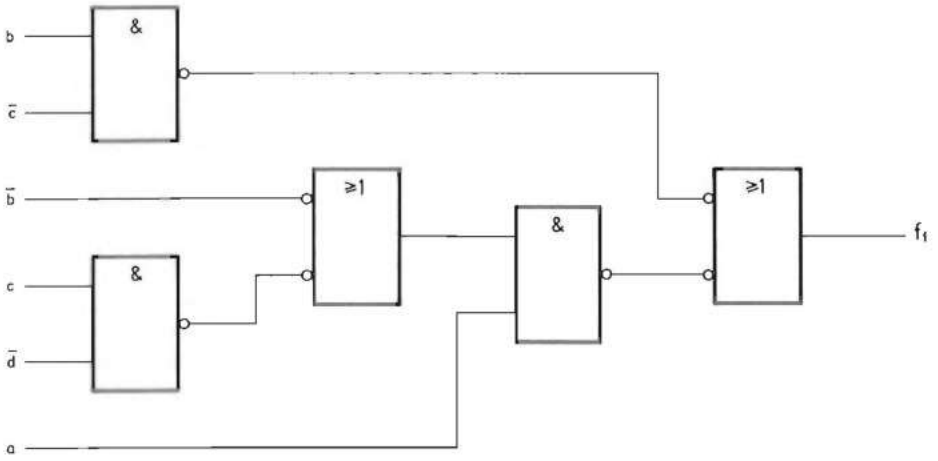


FIGURA 3.33.—**Esquema equivalente al de la figura 3.31.**

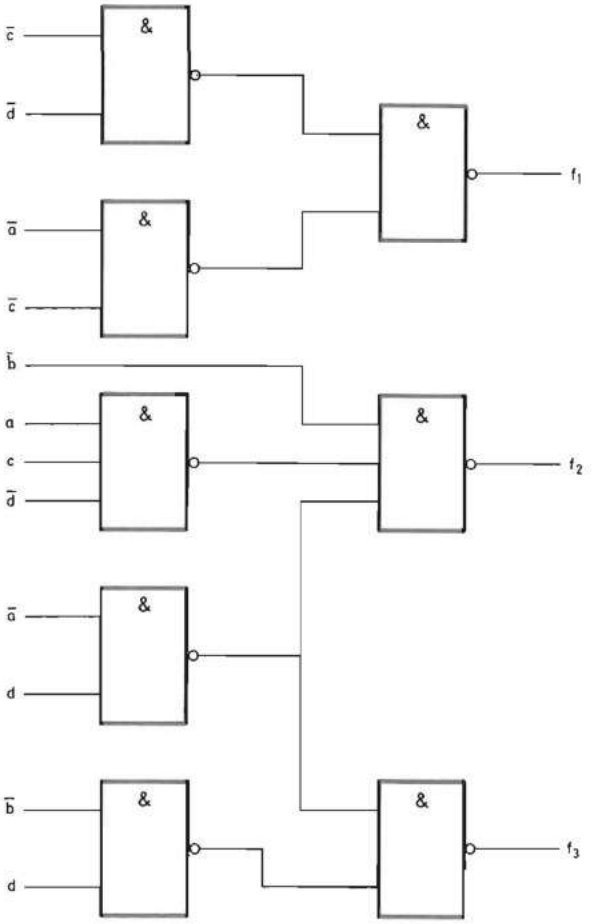


FIGURA 3.34. —**Esquema con puertas NO-Y (NAND) de la multifunción** $f_1 = \bar{c}\,\bar{d} + \bar{a}\,\bar{c}$, $f_2 = b + a\,c\,\bar{d} + \bar{a}\,d$ **y** $f_3 = \bar{b}\,d + \bar{a}\,d$.

Comparando la figura 3.15 con la 3.30 se observa que en esta última se utiliza el mismo número de puertas que en aquélla, con una entrada menos, pero que su nivel es 3 en lugar de 2.

Las multifunciones se realizan de la misma forma con puertas NO-Y y NO-O pero realizando una sola vez los términos comunes. Como ejemplo se realiza con puertas NO-Y la multifunción diseñada en el apartado 3.4.

Aplicando las reglas antes indicadas resulta:

$$f_1 = \bar{c}\bar{d} + \bar{a}\bar{c} = \overline{\overline{\bar{c}\bar{d}} + \overline{\bar{a}\bar{c}}} = \overline{\overline{\bar{c}\bar{d}}\ \ \overline{\bar{a}\bar{c}}}$$

$$f_2 = b + ac\bar{d} + \bar{a}d = \overline{\overline{b + ac\bar{d} + \bar{a}d}} = \overline{\bar{b}\ \ \overline{ac\bar{d}}\ \ \overline{\bar{a}d}}$$

$$f_3 = \bar{b}d + \bar{a}d = \overline{\overline{\bar{b}d} + \overline{\bar{a}d}} = \overline{\overline{\bar{b}d}\ \ \overline{\bar{a}d}}$$

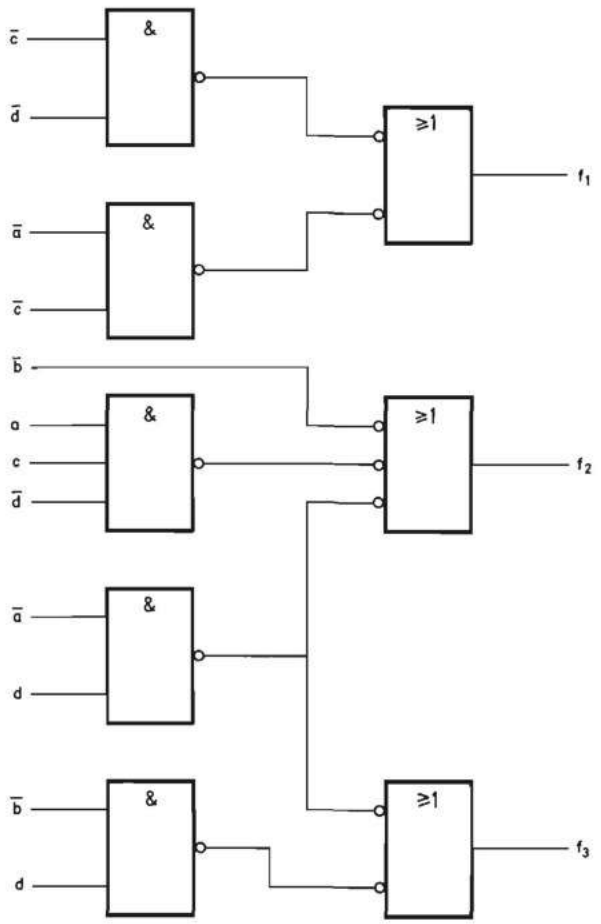


FIGURA 3.35.—**Esquema equivalente al de la figura 3.34.**

En la figura 3.34 se representa el circuito correspondiente. Aplicando el criterio de representación explicado en párrafos anteriores se obtiene el esquema de la figura 3.35, equivalente al de la 3.34.

3.5.2 Realización de funciones mediante el montaje «Y por conexión»

En las tecnologías en las que el transistor de salida posee como carga una resistencia [tecnología DTL y algunos circuitos de la TTL (ver apartado 5.4.4.3.2 del capítulo 5)] se pueden unir directamente las salidas de las puertas NO-Y (NAND) obteniéndose la función denominada «Y por conexión» (en inglés «Wired And»). La figura 3.36 representa este montaje con los símbolos antiguos y la 3.37 con los nuevos nor-

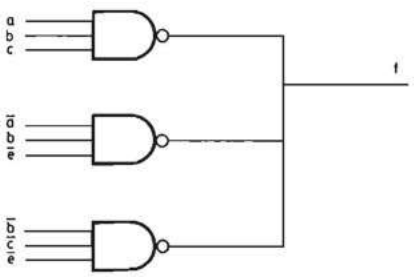


FIGURA 3.36.—**Ejemplo del montaje «Y por conexión» con símbolos no normalizados.**

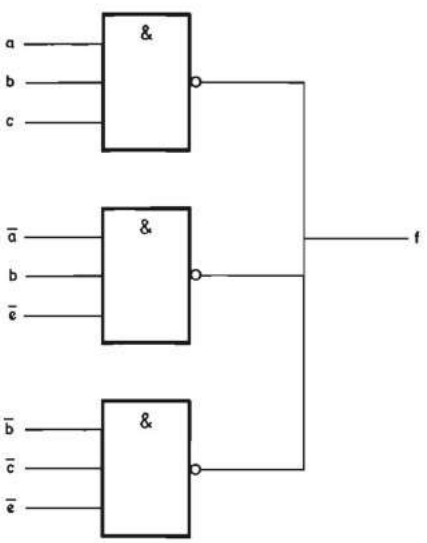


FIGURA 3.37.—**Ejemplo del montaje «Y por conexión» con símbolos normalizados.**

malizados. La función f toma el valor cero cuando todas las entradas de cualquiera de las puertas NO-Y se encuentran en estado uno simultáneamente. Por lo tanto:

$$f = \overline{abc}\ \overline{\bar{a}b\bar{e}}\ \overline{\bar{b}ce}$$

Una de las ventajas del nuevo sistema de representación gráfica de las funciones lógicas es la adopción de símbolos para las distintas configuraciones de las etapas de salida descritas en el apartado A1.3 del apéndice 1. De acuerdo con la nueva simbología el esquema de la figura 3.37 debe ser sustituido por el de la figura 3.38.

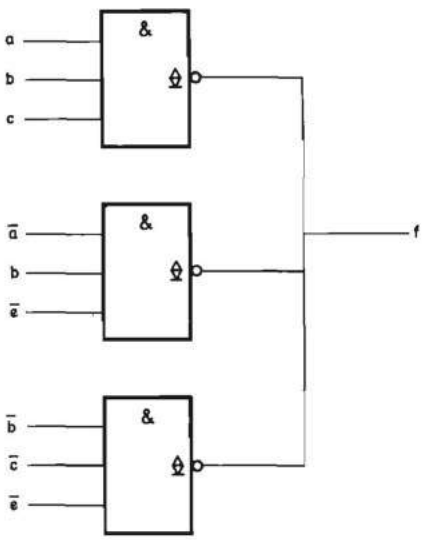


FIGURA 3.38. —**Ejemplo del montaje «Y por conexión» con símbolos normalizados con indicativo de salida con carga pasiva en el colector.**

El montaje «Y por conexión» se puede realizar también mediante puertas de colector abierto, que reciben esta denominación porque el transistor de salida carece de carga (ver la figura 5.36 del capítulo 5). El esquema, en este caso, del circuito que realiza la función f se representa en la figura 3.39. Cualquiera de las expresiones mínimas de producto de sumas o suma de productos puede ser realizada mediante el montaje «Y por conexión». En algunos casos, la realización de las funciones se simplifica mucho mediante la utilización de este montaje.

El progreso de las técnicas de integración ha disminuido el interés de la utilización del montaje «Y por conexión» en la generación de funciones por obtenerse tiempos de propagación y consumos superiores a los de las puertas con carga activa (ver apartado 5.4.4.3.2 del capítulo 5). No obstante, este montaje se utiliza en circuitos de control de entrada-salida de los microprocesadores y por ello para el lector es interesante su comprensión.

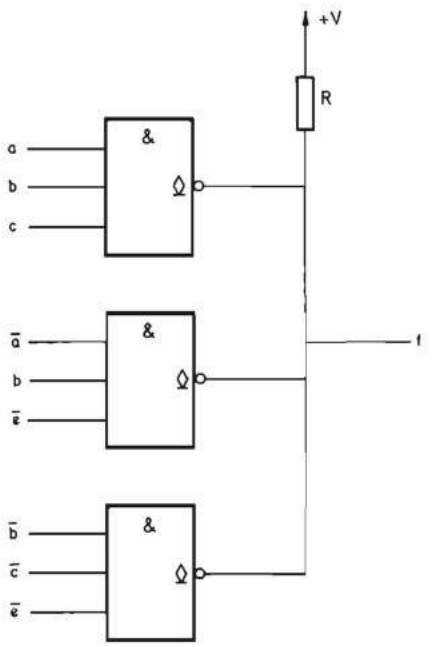


FIGURA 3.39.—**Ejemplo del montaje «Y por conexión» realizado con puertas de colector abierto.**

3.5.3 Realización de las funciones lógicas con puertas O-exclusiva

La utilización de puertas O-exclusiva de dos entradas disponibles en circuito integrado de pequeña escala de integración, en combinación con puertas NO-Y y NO-O, permite reducir el número total de puertas necesario para la realización de ciertas funciones lógicas. Este hecho tenía gran interés cuando la microelectrónica no había alcanzado el nivel de gran escala de integración (ver apartado 5.4.4.2.1.2 del capítulo 5). En esa época (1965-1975) se desarrolló un método tabular de síntesis de funciones lógicas mediante la función O-exclusiva que constituyó una importante herramienta de trabajo en el diseño de sistemas digitales. Al alcanzarse los niveles de gran y muy gran escala de integración y no ser necesaria la simplificación máxima de los circuitos combinacionales, este método ha perdido una gran parte de su utilidad y por ello no se incluye en esta edición. Al lector interesado en conocerlo se le sugiere que se ponga en contacto con el autor que le facilitará un ejemplar del mismo.

3.6 FENOMENOS ALEATORIOS EN LOS SISTEMAS COMBINACIONALES

En el estudio que se ha realizado en los apartados anteriores no se ha tenido en cuenta el retardo inherente a la propagación de la señal a través de las puertas lógicas. Este retardo se presenta en todas las tecnologías y su magnitud depende en gran medida de aquélla.

A continuación se estudian los tipos más importantes de fenómenos aleatorios en los circuitos combinacionales y la forma de diseñar éstos para evitar su aparición.

3.6.1 Fenómenos aleatorios estáticos

Se presenta un fenómeno aleatorio estático, si para dos estados de las variables de entrada adyacentes en los que la salida debe ser constante (0 o 1), existe un régimen transitorio durante el cual la salida cambia eventualmente de estado. Existen por lo tanto un fenómeno aleatorio de tipo cero y un fenómeno aleatorio de tipo uno.

Los fenómenos aleatorios de tipo cero se producen cuando, al cambiar de estado una variable, la función debe de permanecer en estado uno, pero pasa transitoriamente por cero. Veamos cómo puede producirse un fenómeno de este tipo. Sea una función f que depende de una serie de variables a, b, c, d,... y designemos por X al conjunto de todas las variables excepto la a. En el capítulo 2 se demuestra que f puede expresarse algebraicamente de la forma siguiente:

$$f = f(a, b, c...) = f(a, X) = a\,f(1, X) + \bar{a}\,f(0, X)$$

Para la combinación de estados X_0 de todas las variables excepto la a en que se verifica:

$$f(1, X_0) = f(0, X_0) = 1$$

resulta

$$f = a + \bar{a}$$

En régimen estático, el estado de la función f es uno lógico, pero debido al retardo en la conmutación, al producirse un cambio de estado de a puede suceder que a y $\bar{a}$ se encuentren en estado cero simultáneamente y que durante un cierto tiempo f tome el valor lógico cero.

En la figura 3.40 se representa esta situación cuando $\bar{a}$ sufre un retraso en la conmutación con respecto a a.

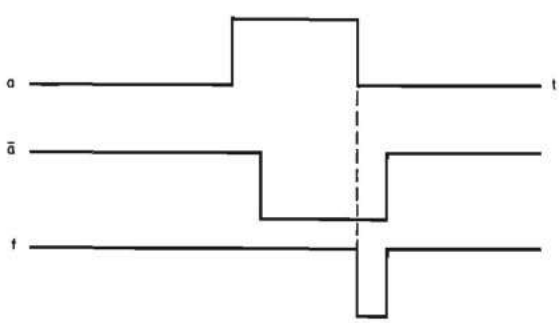


FIGURA 3.40. —**Representación gráfica de un fenómeno aleatorio estático de tipo cero.**

La forma de eliminar este fenómeno es añadir a la expresión de f un término que tome el valor lógico uno para la combinación X_0 que es precisamente el producto lógico $f(1, X)\, f(0, X)$. Aplicando los postulados del álgebra de Boole se demuestra que la expresión de la función f no varía al añadir dicho término, que cubre la adyacencia existente entre los términos $af(1, X)$ y $\bar{a}f(0, X)$. Un ejemplo sencillo lo aclarará.

Sea la función de tres variables

$$f = \sum_3 (3, 4, 6, 7)$$

cuya tabla de Karnaugh se representa en la figura 3.41.

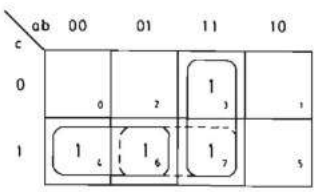


FIGURA 3.41.—**Tabla de Karnaugh de la función** $f = \sum_3 (3,4,6,7)$.

Minimizando la función por el método indicado en el apartado 3.2.1 resulta la expresión:

$$f = ab + \bar{a}c$$

que presenta la posibilidad de la aparición de un impulso aleatorio cuando $b = c = 1$.

En la figura 3.42*a* se representa la realización física de esta función con relés y se observa que, si b y c están cerrados, se produce una apertura transitoria del circuito al actuar el conmutador a si éste abre un contacto antes de cerrar el otro.

Esta función se puede realizar también mediante puertas NO-Y (NAND) transformando adecuadamente su expresión.

$$f = ab + \bar{a}c = \overline{\overline{ab}\ \ \overline{\bar{a}c}}$$

En la figura 3.42*b* se representa esta realización. Si la puerta N1 posee un tiempo de retraso a la propagación superior a N2, se producirá un impulso aleatorio cuando la variable a conmute del estado cero al uno siendo $b = c = 1$.

Para evitar la aparición de este fenómeno aleatorio se ha de sumar lógicamente a la expresión de f el término bc indicado en línea de puntos en la tabla de la figura 3.41 y que cubre la adyacencia entre los otros dos términos. Por tanto, para eliminar los fenómenos aleatorios estáticos de tipo cero al realizar la minimización de una función lógica mediante la tabla de Karnaugh de la suma de productos canónicos es necesario, no sólo cubrir todos los unos de la función, sino también cubrir todas las adyacencias, solapando todos los términos primos.

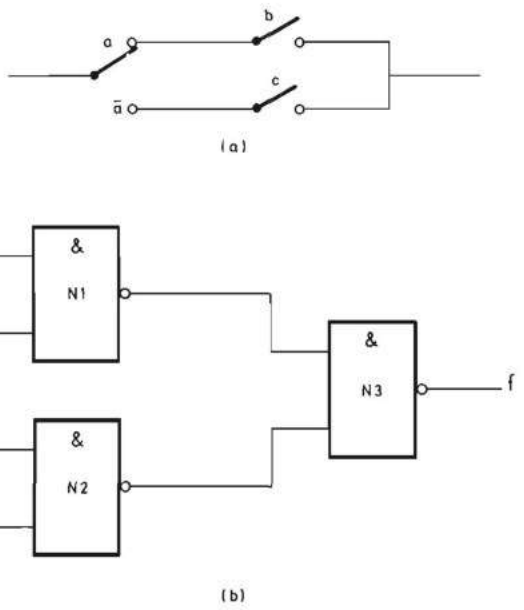


FIGURA 3.42.—**Realización física de la expresión mínima de la función *f* de la figura 3.41 que presenta un fenómeno aleatorio estático de tipo cero.**

La expresión resultante de f es pues

$$f = ab + \bar{a}c + bc$$

cuya realización física se representa en las figuras 3.43*a* y 3.43*b*.

El lector puede comprobar la ausencia en estos circuitos del fenómeno aleatorio anteriormente descrito.

En efecto, en la figura 3.43*b* se observa que, cuando las variables b y c se encuentran en estado uno simultáneamente, la salida de la puerta NO-Y N3 toma el valor lógico cero y, por tanto, la salida de N4 adopta el estado lógico uno independientemente del estado lógico de las restantes entradas.

Los fenómenos aleatorios de tipo uno se producen cuando, al cambiar de estado una variable, la función debe de permanecer en estado cero, pero pasa transitoriamente por el estado uno. Este fenómeno puede producirse cuando una función ha sido expresada mediante un producto de sumas. Sea una función f de las variables a, b, c, d, ..., que puede expresarse, tal como vimos en el capítulo 2, de la forma siguiente:

$$f(a, b, c, d...) = f(a, X) =$$
$$= [a + f(0, X)]\,[\bar{a} + f(1, X)]$$

Si para la combinación de estados X_0 de todas las variables excepto la a se verifica:

$$f(0, X_0) = f(1, X_0) = 0$$

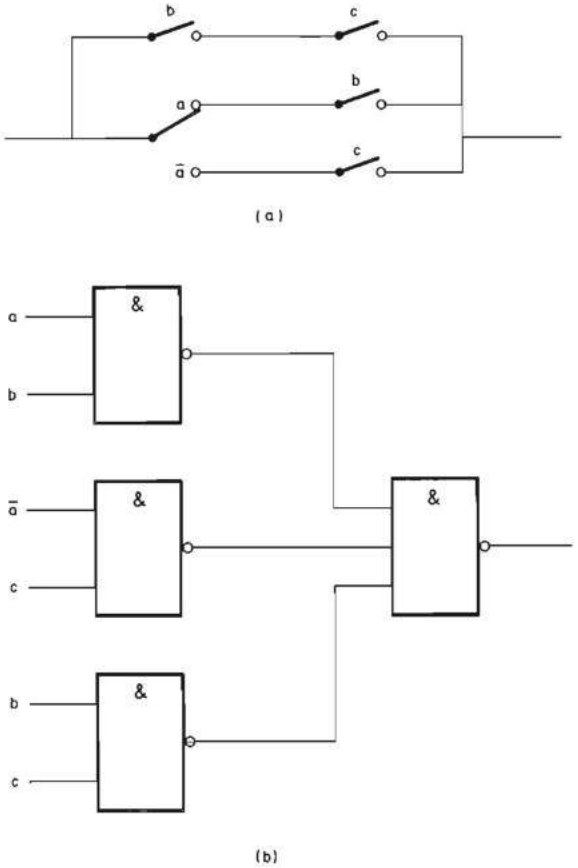


FIGURA 3.43.—**Realización física de la función f de la figura 3.41 en la que se ha suprimido el fenómeno aleatorio estático de tipo cero.**

resulta:

$$f = a\bar{a}$$

En régimen estático la función f se encuentra en estado lógico cero, pero al producirse un cambio en la variable a debido al retardo en la conmutación, puede suceder que a y $\bar{a}$ tomen el valor uno simultáneamente y que durante un cierto tiempo f tome por tanto el valor lógico uno. En la figura 3.44 se representa gráficamente este fenómeno.

La forma de eliminar este fenómeno es similar a la descrita para el tipo cero. La expresión de f ha de multiplicarse por $f(0, X) + f(1, X)$ que toma el valor lógico cero para la combinación X_0 de entrada. También se demuestra que la expresión de la función f no varía al añadir dicho término que cubre la adyacencia existente entre $a + f(0, X)$ y $\bar{a} + f(1, X)$.

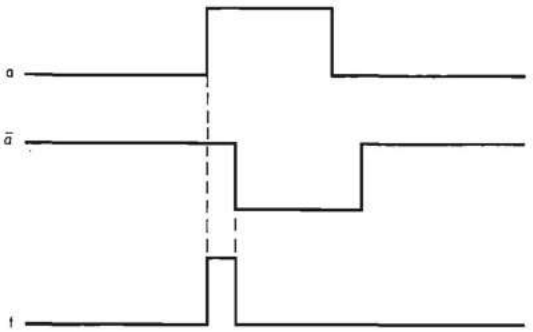


FIGURA 3.44.—**Representación gráfica de un fenómeno aleatorio estático de tipo uno.**

Veremos también un ejemplo sencillo. Sea la función f de tres variables,

$$f = \prod_3 (2, 3, 5, 7)$$

cuya tabla de Karnaugh se representa en la figura 3.45.

La expresión mínima deducida de dicha tabla es,

$$f = (b + \bar{c})\,(a + c)$$

Si $b = a = 0$ resulta:

$$f = c\,\bar{c}$$

y se puede presentar un fenómeno aleatorio de tipo uno como puede verse en los esquemas de realización física con relés y puertas NO-O de la figura 3.46.

Para eliminar este fenómeno aleatorio se ha de multiplicar lógicamente la expresión anterior por el término $a + b$ que se ha indicado en línea de puntos en la tabla de la figura 3.45. Se deduce como regla que, para minimizar mediante el método tabular de Karnaugh una función lógica expresada como producto de sumas canónicas, se han de cubrir todas las adyacencias, solapando de esta forma todos los términos primos.

La expresión resultante de f es:

$$f = (b + \bar{c})\,(a + c)\,(a + b)$$

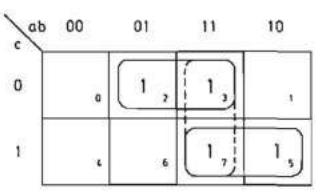


FIGURA 3.45.—**Tabla de Karnaugh de la función** $f = \prod_3 (2, 3, 5, 7)$.

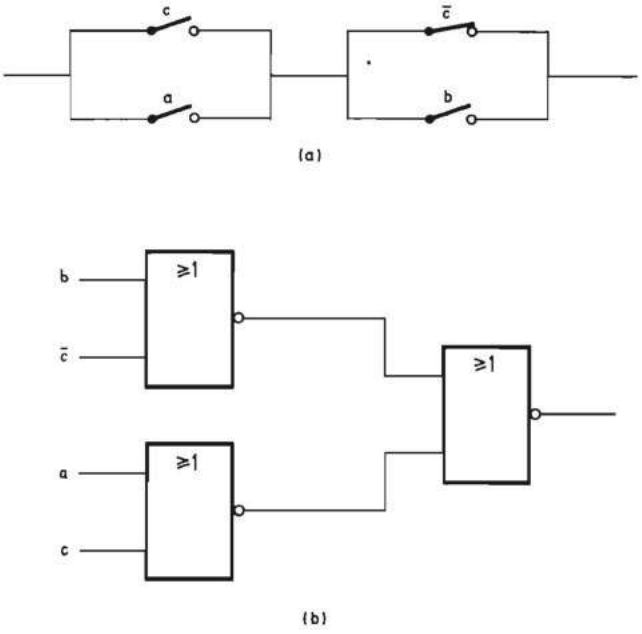


FIGURA 3.46.—**Realización de la expresión mínima de la función f de la figura 3.45 que presenta un fenómeno aleatorio estático de tipo uno.**

que, como puede observarse en la figura 3.47 que representa su realización física, está exenta de fenómenos aleatorios del tipo descrito.

Si se desea realizar la minimización de una función lógica por el método numérico, se han de cubrir igualmente todas las adyacencias a fin de evitar la aparición de fenómenos aleatorios estáticos de tipo cero o uno.

Esto se logra haciendo que los términos primos obtenidos por el procedimiento explicado en el apartado 3.2.2 realicen todos los términos canónicos que no tienen ningún otro adyacente y las agrupaciones de dos términos canónicos.

Como ejemplo minimizaremos la función

$$f = \sum_4 (1, 2, 3, 5, 6, 7, 9, 10, 11, 14)$$

En la tabla 3.15 se representan las tablas de agrupación de los términos canónicos obtenidas por el procedimiento indicado en el apartado 3.2.2.

En la tabla 3.16 se presenta la relación de términos primos cuyas columnas corresponden a los términos de la tabla 2 de la 3.15. Se comprueba que todos los términos primos son esenciales y, por tanto, la expresión mínima de f en forma de suma de productos es:

$$f = a\bar{d} + a\bar{c} + b\bar{d} + b\bar{c} + \bar{a}b$$

El lector puede comprobar que, sin tener en cuenta los fenómenos aleatorios, serían suficientes los términos $a\bar{c}$, $\bar{a}b$ y $a\bar{d}$.

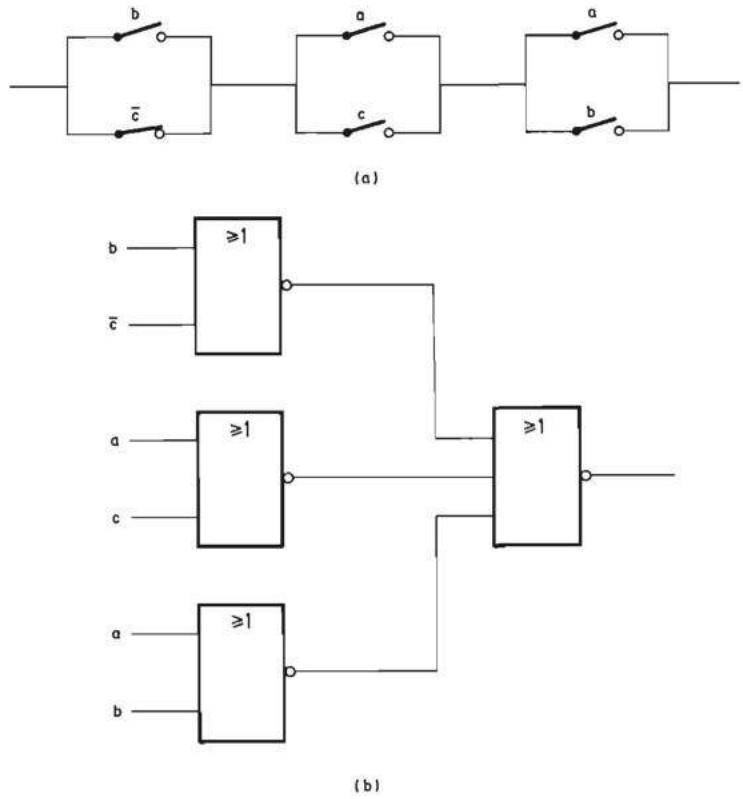


FIGURA 3.47.—**Realización física de la función *f* de la figura 3.45 en la que se ha suprimido el fenómeno aleatorio estático de tipo uno.**

3.6.2 Fenómenos aleatorios dinámicos

Se produce un fenómeno aleatorio dinámico cuando, al conmutar una variable de entrada, la salida debe cambiar de estado y lo hace pasando por un régimen transitorio tal como se indica en la figura 3.48.

Se analizará a continuación este fenómeno para las expresiones en forma de suma de productos. Para que un fenómeno aleatorio de tipo dinámico pueda producirse es necesario que, para una combinación determinada de valores lógicos de todas las variables de entrada, excepto la a, la salida del circuito adopte la expresión mínima:

$$f = a + a \quad \bar{a}$$

En régimen estático $f = a$, pero veamos lo que puede ocurrir al cambiar a de 0 a 1. Supondremos que el cambio de estado de todas las variables a no es simultáneo y las denominaremos a_1, a_2 y a_3 y por tanto:

$$f = a_1 + a_2 \quad \bar{a}_3$$

Tabla 1

N.º de unos		
1	1	X
	2	X
	3	X
	5	X
2	6	X
	9	X
	10	X
	7	X
3	11	X
	14	X

Tabla 2

			Dif.
1 - 3	X	2	
1 - 5	X	4	
1 - 9	X	8	
2 - 3	X	1	
2 - 6	X	4	
2 -10	X	8	
3 - 7	X	4	
3 -11	X	8	
5 - 7	X	2	
6 - 7	X	1	
6 -14	X	8	
9 -11	X	2	
10 -11	X	1	
10 -14	X	4	

Tabla 3

	Dif.
1-3-5-7	2,4
1-3-9-11	2,8
1-5-3-7	4,2
1-9-3-11	8,2
2-3-6-7	1,4
2-3-10-11	1,8
2-6-3-7	4,1
2-6-10-14	4,8
2-10-3-11	8,1
2-10-6-14	8,4

Tabla 3 *reducida*

	Dif.
1-3-5-7	2,4
1-3-9-11	2,8
2-3-6-7	1,4
2-3-10-11	1,8
2-6-10-14	4,8

TABLA 3.15

Términos primos	1-3	1-5	1-9	2-3	2-6	2-10	3-7	3-11	5-7	6-7	6-14	9-11	10-11	10-14
1-3-5-7	X	X					X		⊗					
1-3-9-11	X		⊗					X				⊗		
2-3-6-7				X	X		X			⊗				
2-3-10-11				X		X		X					⊗	
2-6-10-14					X	X					⊗			⊗

TABLA 3.16

Supongamos también que $a_1 = a_2 = 1$, $\bar{a}_3 = 0$ y por tanto $f = 1$. Si a cambia del estado uno al cero, se produce un fenómeno aleatorio dinámico cuando los cambios de a_1, a_2 y $\bar{a}_3$ se realizan de acuerdo con la siguiente tabla

	a_1	a_2	$\bar{a}_3$	f
Estado inicial	1	1	0	1
Estados	0	1	0	0
Transitorios	0	1	1	1
Estado final	0	0	1	0

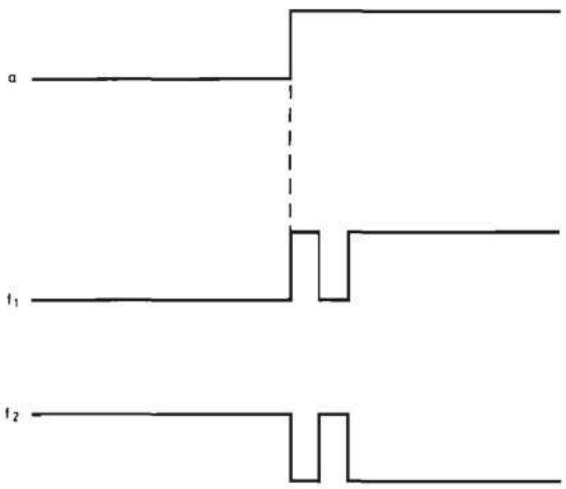


FIGURA 3.48.—**Representación gráfica de un fenómeno aleatorio dinámico.**

Se comprueba la igualdad de este cambio con el descrito para la función f_2 de la figura 3.48.

Un ejemplo práctico es el circuito de la figura 3.49 que corresponde a la función:

$$f = \overline{\overline{\overline{ab}\ \ \overline{\bar{a}\bar{c}}\ \ \overline{ad}}} = \overline{ab}\ \ \overline{\bar{a}\bar{c}} + ad = [(\bar{a} + \bar{b})\,(a + c)] + ad$$

Si

$$b = d = 1 \text{ y } c = 0$$

Se verifica

$$f = \bar{a}a + a$$

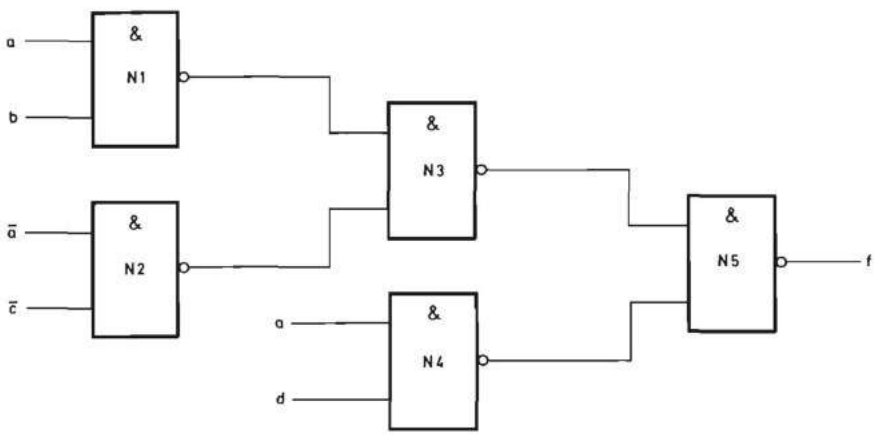


FIGURA 3.49.—**Ejemplo de circuito que presenta un fenómeno aleatorio dinámico.**

y observando la figura 3.49 se deduce que para que se produzca un fenómeno aleatorio dinámico al cambiar a de uno a cero es necesario que el tiempo de propagación de la señal a través de N_4 sea inferior al de N_1 más el de N_3 y al de N_2 más el de N_3 y además que a su vez el tiempo de propagación de N_2 sea menor al de N_1.

La eliminación de la posibilidad de fenómenos aleatorios dinámicos se realiza al minimizar una función por el método de Karnaugh o el numérico indicados en los apartados 3.2.1 y 3.2.2 porque la expresión de suma de productos obtenida nunca contendrá el producto $a\bar{a}$ (o la suma $a + \bar{a}$ si se trata de una expresión de producto de sumas). Por tanto, si además se realiza la minimización anulando la posibilidad de fenómenos aleatorios estáticos, se tiene la seguridad de que no existirán fenómenos aleatorios dinámicos.

3.7 BLOQUES FUNCIONALES COMBINACIONALES

En apartados anteriores se han estudiado las funciones lógicas y su realización mediante puertas NO-Y (NAND) y NO-O (NOR). Existen funciones lógicas y, en especial, multifunciones que presentan la característica de ser de aplicación general, es decir, que pueden ser utilizadas como bloques funcionales para la construcción de sistemas digitales.

El progreso de las técnicas de integración permitió (en la década de los sesenta) la realización de dichas funciones y multifunciones en circuito integrado, constituyendo lo que se ha denominado la escala de integración media (MSI). La combinación de estos circuitos con los secuenciales estudiados en el capítulo 6 generó bloques funcionales complejos que al ser integrados han dado lugar a los circuitos de gran (LSI), muy gran (VLSI) y ultra gran (ULSI) escala de integración.

Los bloques funcionales combinacionales responden, en general, al diagrama de bloques de la figura 3.50 en el que las variables de entrada se dividen en dos grupos:

a) Variables de entrada con las que el circuito realiza una determinada función u operación y a las que se puede denominar operativas.
b) Variables de entrada que influyen en la forma en que el circuito actúa sobre las operativas o que modifican el resultado o variables de salida del propio circuito. Reciben el nombre de variables de control.

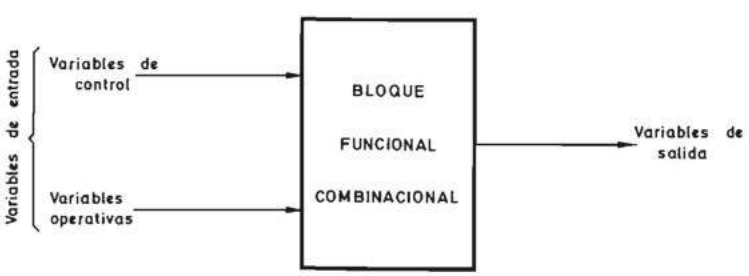


FIGURA 3.50.—**Diagrama de bloques de un bloque funcional combinacional.**

Se dice que existe entre ambos tipos de entradas una relación de dependencia. Hay varios tipos de dependencias que se distinguen por la forma en que se realiza la citada modificación, pero existe una regla general, común a todas, que es la siguiente:

La entrada que modifica la forma en que el circuito actúa sobre las de información u operativas o que modifica el resultado se indica con una letra correspondiente al tipo de dependencia seguida de un número. Cada una de las entradas afectadas tiene asignado el mismo número.

Las principales variables de control de los bloques funcionales combinacionales y las relaciones de dependencia asociadas a ellas son:

— Entrada o entradas de desinhibición/inhibición (enable/disable) que como su nombre indica inhiben o desinhiben la acción del circuito sobre las demás entradas y ponen las salidas en un determinado nivel lógico. La relación de dependencia correspondiente puede ser de tipo Y [G (AND)] o de tipo O [V (OR)]. En el primer caso, la colocación en cero de la entrada de control pone a cero las variables de salida y en el segundo, la colocación en uno de la entrada de control pone a uno las salidas.
— Entrada o entradas de control del tercer estado de las variables de salida. Cuando estas variables se encuentran en un determinado estado lógico, la salida se pone en tercer estado, que recibe esta denominación porque en él la salida presenta una gran impedancia a ambos polos de la tensión de alimentación. En el capítulo 5 se estudian las puertas de tres estados en diferentes tecnologías. Esta relación de dependencia recibe el nombre de desinhibición [EN (Enable)].
— Entradas de control de inversión de las variables operativas, que según cual sea su estado lógico hacen que el circuito actúe directamente sobre las variables operativas o sobre sus inversas. Esta relación de dependencia recibe el nombre de inversión [N (Negate)].
— Entradas de control de inversión de las variables de salida, que según cual sea su estado lógico hacen que a la salida aparezcan las variables en forma directa o invertida. Esta relación de dependencia recibe también el nombre de inversión [N (Negate)].
— Entradas de selección de operación o de modo de operación que como su nombre indica seleccionan la operación que realiza el circuito con las entradas operativas. La relación de dependencia correspondiente recibe el nombre de «modo de operación» [M (Mode)].

En este apartado se estudian los bloques funcionales combinacionales cuyo conocimiento resulta imprescindible para diseñar sistemas digitales y comprender el funcionamiento de los bloques más complejos. En su representación gráfica se utilizan los símbolos lógicos normalizados que se describen en el apartado A1.4.3. del apéndice 1.

3.7.1 Decodificadores. Demultiplexores

a) Decodificadores.

Los circuitos decodificadores son sistemas combinacionales que generan los

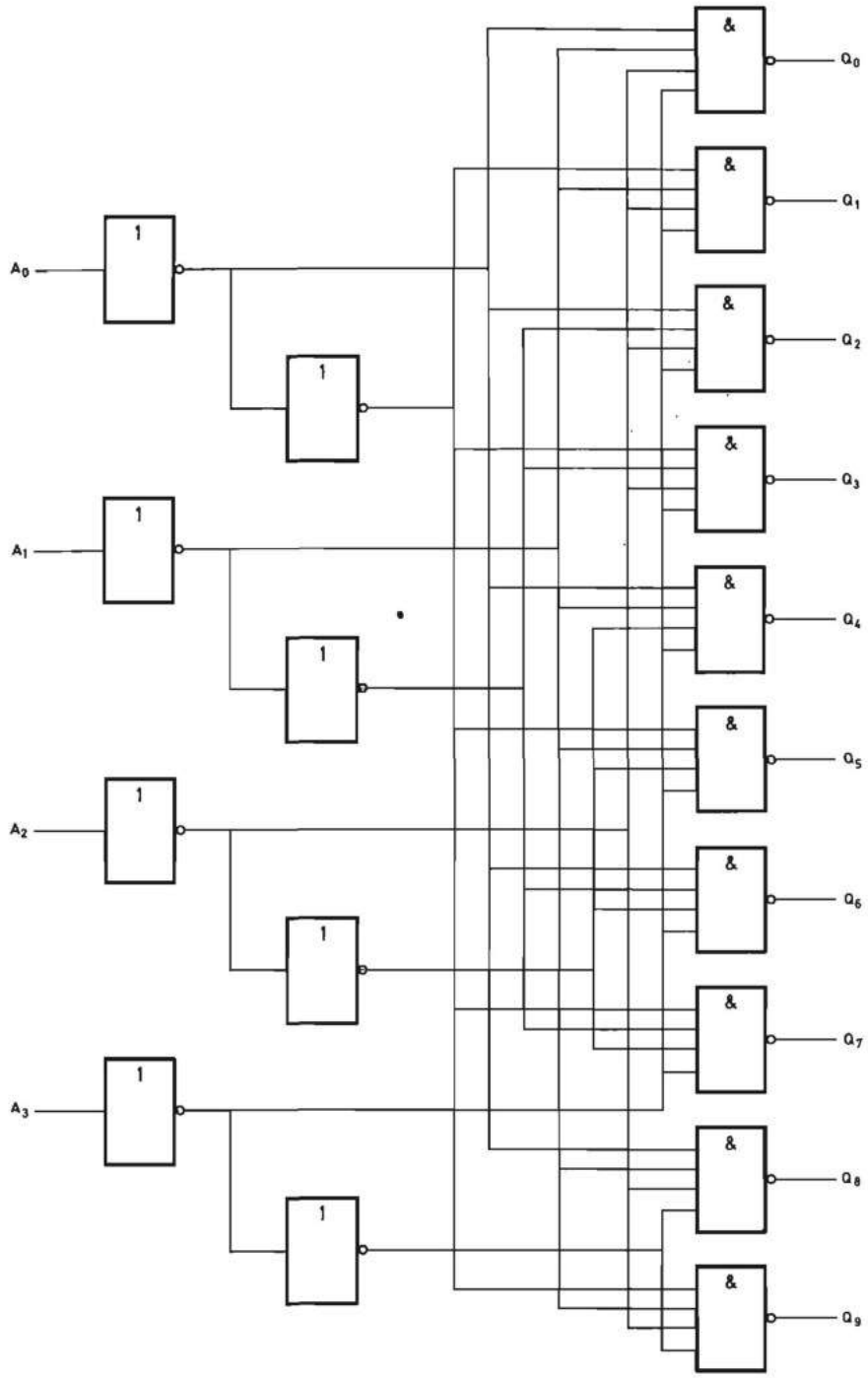


FIGURA 3.51.—**Esquema de un decodificador decimal.**

productos canónicos de un conjunto de variables binarias aplicadas a sus entradas y se dividen en excitadores (drivers) y no excitadores, según que sus salidas puedan o no controlar respectivamente a un indicador numérico. También reciben el nombre de decodificadores excitadores algunos circuitos que son realmente convertidores de código (ver figura 3.65).

En primer lugar se estudian los decodificadores no excitadores que poseen innumerables aplicaciones.

En la figura 3.51 se representa el esquema de un decodificador decimal que genera los productos canónicos 0 a 9 de cuatro variables binarias. Una cualquiera de las salidas toma el valor cero lógico solamente cuando aparece una determinada combinación de entrada. En la tabla 3.17 se representa la tabla de verdad del decodificador decimal, que el lector puede comprobar analizando el funcionamiento del circuito de la figura 3.51. La doble inversión de las variables A_0 a A_3 tiene como finalidad hacer que el decodificador cargue al circuito que se conecte a su entrada con el equivalente a una puerta lógica.

En la figura 3.52 se representa el símbolo lógico de este circuito en versión antigua y en la nueva normalizada. La aplicación inmediata de este circuito es la conversión del código BCD natural al decimal. En efecto aplicando a las entradas 1, 2, 4 y 8 (A_0 a A_3 en el símbolo no normalizado) una combinación del código BCD natural, solamente una de las salidas toma el valor lógico cero y a dicha salida se le asigna el número decimal equivalente a dicha combinación.

Entradas				*Salidas*									
A_3	A_2	A_1	A_0	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

TABLA 3.17.—**Tabla de verdad del decodificador decimal.**

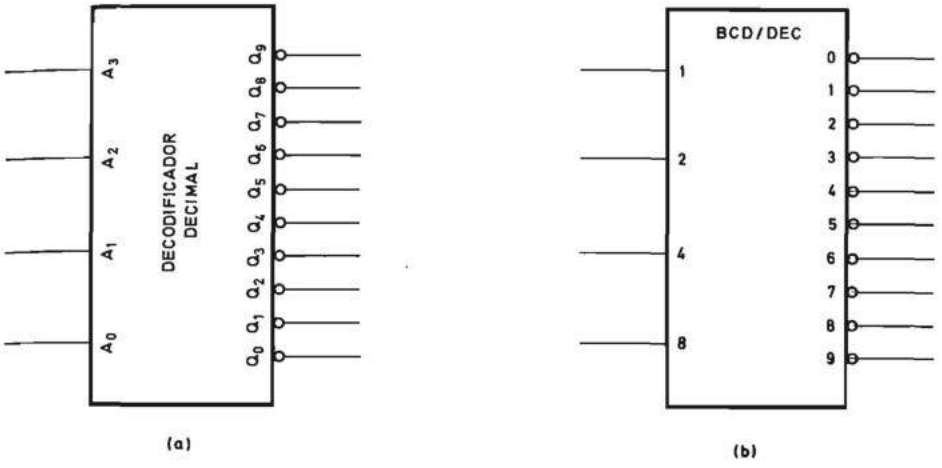


FIGURA 3.52.—**Símbolo lógico de un decodificador decimal: *a*) no normalizado; *b*) normalizado.**

I_1	I_2	A_3	A_2	A_1	A_0	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_{11}	Q_{12}	Q_{13}	Q_{14}	Q_{15}
1	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

X Las entradas correspondientes pueden tomar los valores 0 o 1 lógicos.

TABLA 3.18.—**Tabla de verdad del decodificador hexadecimal.**

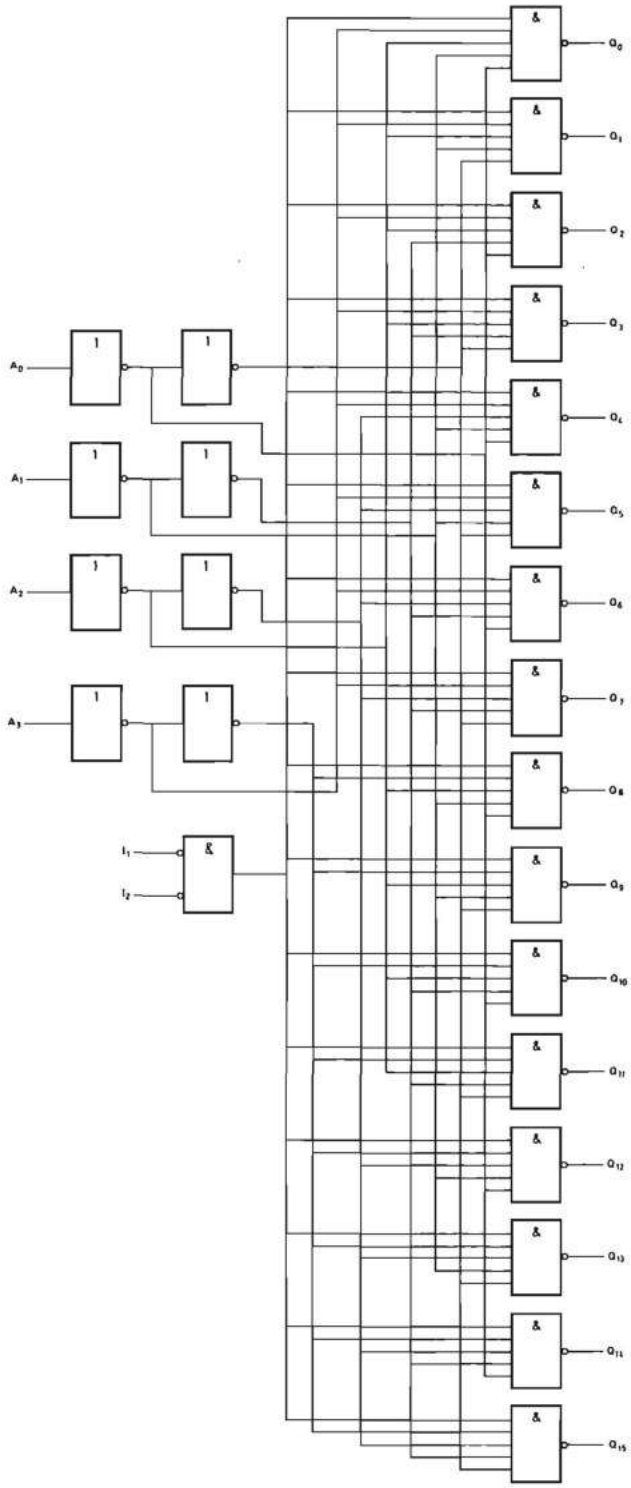


FIGURA 3.53. —**Esquema de un decodificador hexadecimal (uno entre dieciséis).**

En la figura 3.53 se representa con la simbología normalizada el esquema de un decodificador que genera los dieciséis productos canónicos de una función de cuatro variables aplicadas a sus entradas A_0 a A_3.

Al igual que el decodificador que se acaba de estudiar, una salida toma el valor cero solamente cuando se presenta una determinada combinación binaria en las entradas A_0 a A_3. Existen, además, dos entradas de inhibición I_1 e I_2; cuando cualquiera de ellas toma el valor lógico uno, todas las salidas toman el valor lógico uno. En la tabla 3.18 se presenta la tabla de verdad de este circuito.

En la figura 3.54 se representa el símbolo normalizado de este decodificador que presenta la ventaja, además, de que su tabla de verdad se deduce directamente de él. Las salidas 0 a 15 corresponden a los 16 productos canónicos de las variables de entrada 1, 2, 4 y 8. La inversión colocada en cada salida indica que en ellas aparece el inverso del producto lógico correspondiente. La denominación de las entradas de control con el apelativo de desinhibición (enable) en lugar de inhibición (disable) hace ver que sólo se produce la decodificación si $\overline{EN_1}$ (I_1) y $\overline{EN_2}$ (I_2) están en nivel cero simultáneamente (función &).

La aplicación más inmediata del decodificador hexadecimal es la generación de todos los productos canónicos de las cuatro variables A, B, C y D. Para ello se conectan éstas a las entradas 1, 2, 4 y 8 y se obtienen los productos canónicos invertidos P_0 a P_{15} en las salidas 0 a 15 (fig. 3.54).

En la figura 3.55 se representa un decodificador de uno entre 32 estados realizado mediante dos decodificadores hexadecimales. Este circuito genera todos los

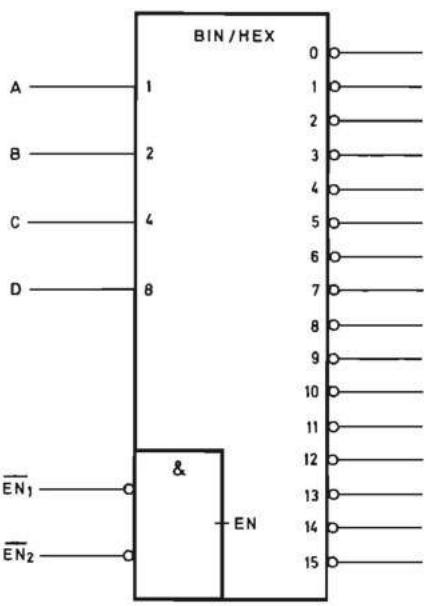


FIGURA 3.54.—**Símbolo lógico normalizado del decodificador hexadecimal de la figura 3.53.**

productos canónicos de 5 variables binarias. Se deja al lector que realice su análisis y compruebe su correcto funcionamiento.

En la figura 3.56 se representa el símbolo lógico de un decodificador de binario

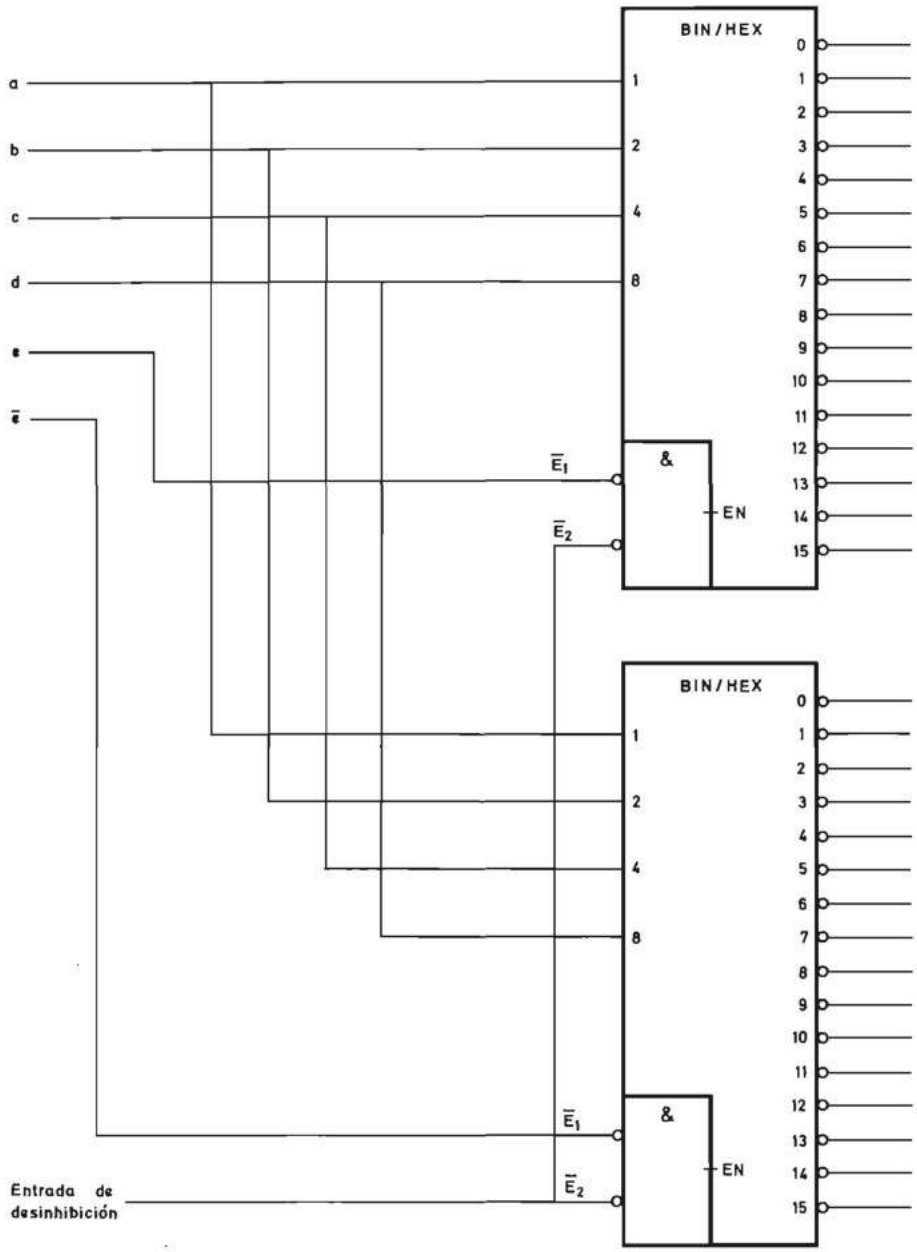


FIGURA 3.55.—**Decodificador de uno entre treinta y dos realizado con dos decodificadores hexadecimales.**

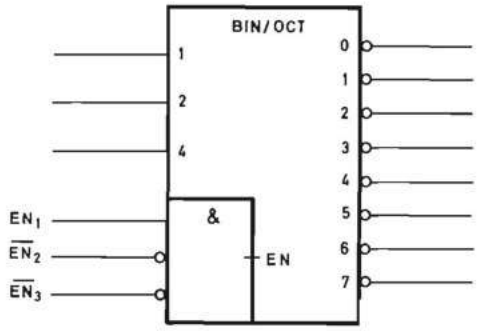


FIGURA 3.56.—**Símbolo lógico normalizado de un decodificador de binario a octal.**

a octal que posee tres entradas de desinhibición que generan una desinhibición interna *EN* por medio de la función:

$$EN = EN_1 \overline{EN_2}\,\overline{EN_3}$$

En la figura 3.57 se representa el esquema de un decodificador de uno entre cuatro que genera todos los productos canónicos de dos variables. La tabla de verdad de este decodificador que posee una entrada de inhibición se representa en la tabla 3.19.

I	A_1	A_0	Q_0	Q_1	Q_2	Q_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

X Las entradas correspondientes pueden tomar los valores 0 o 1 lógicos

TABLA 3.19

En la figura 3.58 se representa el símbolo lógico normalizado de un circuito de escala de integración media que contiene dos decodificadores de uno entre cuatro.

La utilización conjunta del decodificador uno entre cuatro y los decodificadores decimal y hexadecimal permite realizar fácilmente decodificadores de un número elevado de variables.

En la figura 3.59 se indica el esquema de un decodificador de seis variables realizado por un decodificador de uno entre cuatro en combinación con cuatro decodificadores hexadecimales.

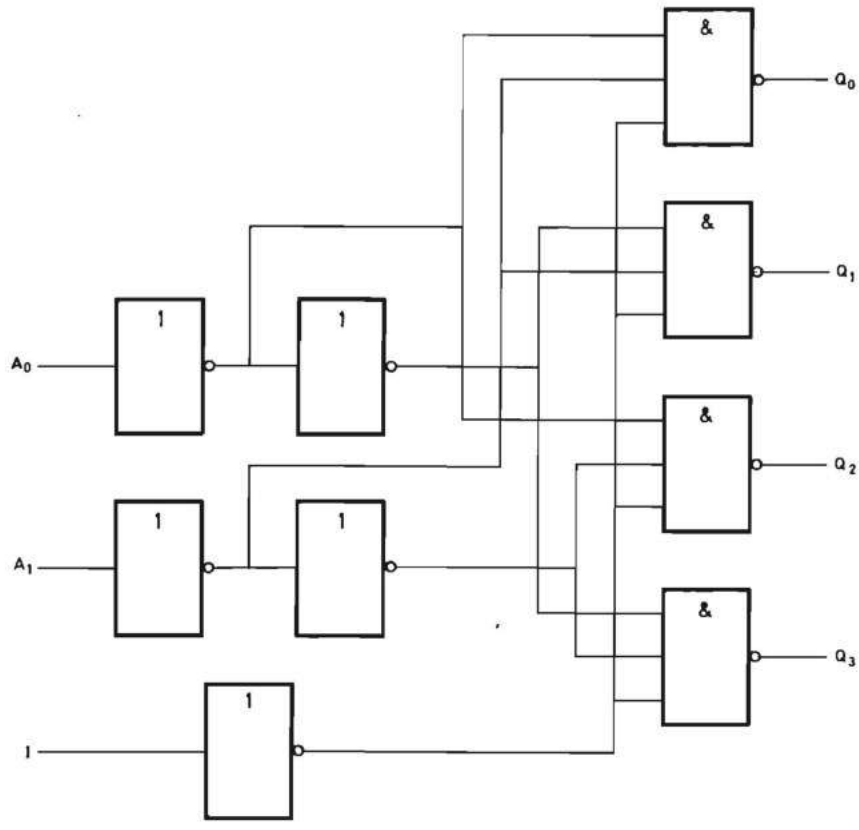


FIGURA 3.57.—**Esquema de un decodificador de uno entre cuatro.**

Los decodificadores admiten diferentes variables de control. Como ejemplo en la figura 3.60 se representa un decodificador de uno entre cuatro que posee tres variables de control:

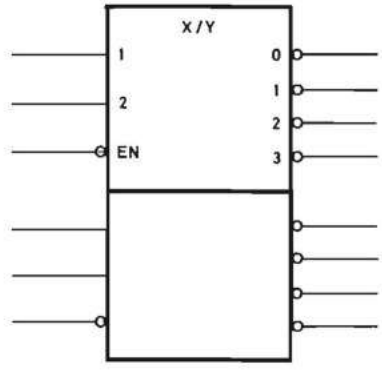


FIGURA 3.58.—**Símbolo lógico normalizado de un doble decodificador de uno entre cuatro.**

— Una variable de desinhibición $\overline{G}$ que desinhibe la decodificación cuando se encuentra en nivel lógico cero. En efecto, si G se encuentra en nivel uno, se ponen a nivel uno las salidas de todas las puertas NO-Y (NAND).

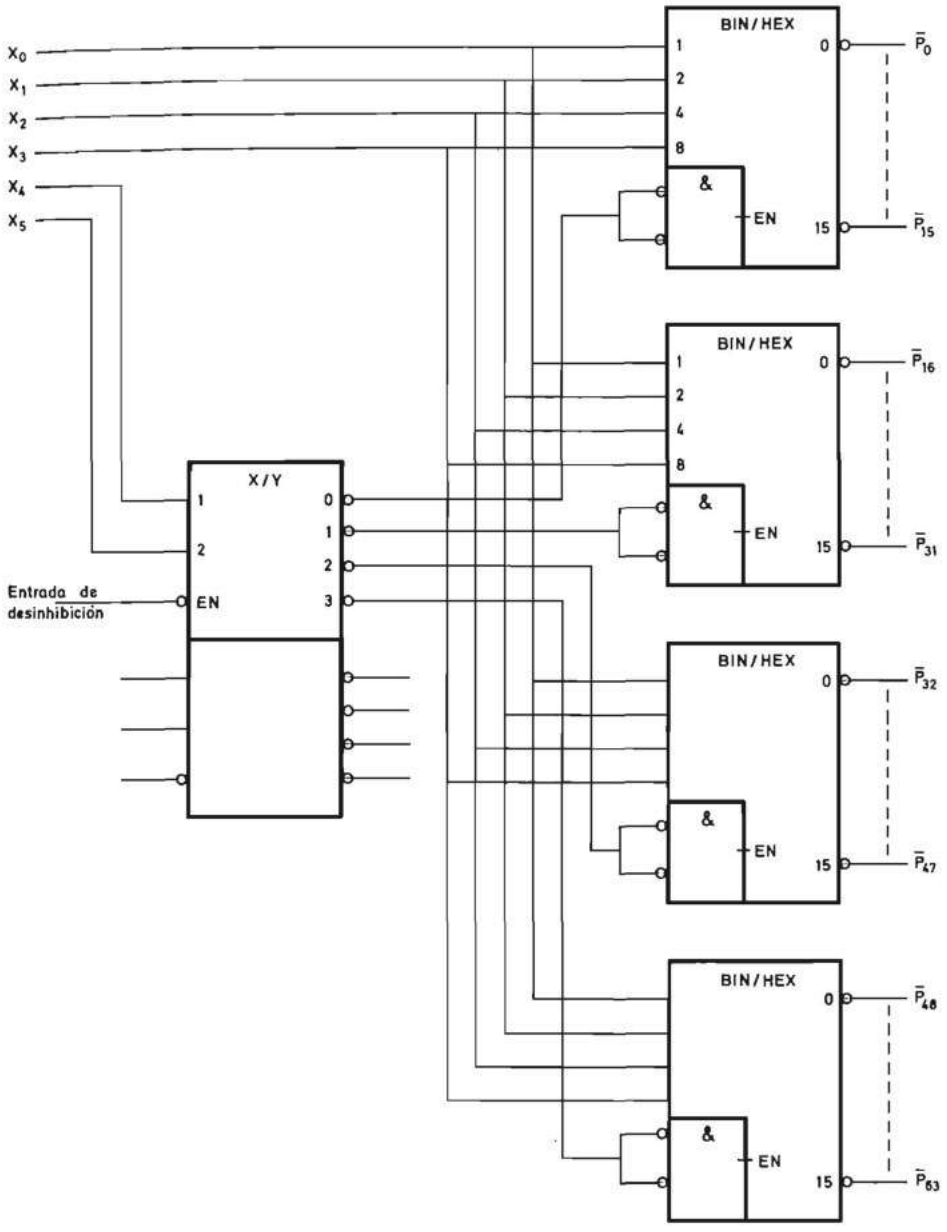


FIGURA 3.59.—**Esquema de un decodificador de seis variables binarias realizado con un decodificador de uno entre cuatro y cuatro decodificadores hexadecimales.**

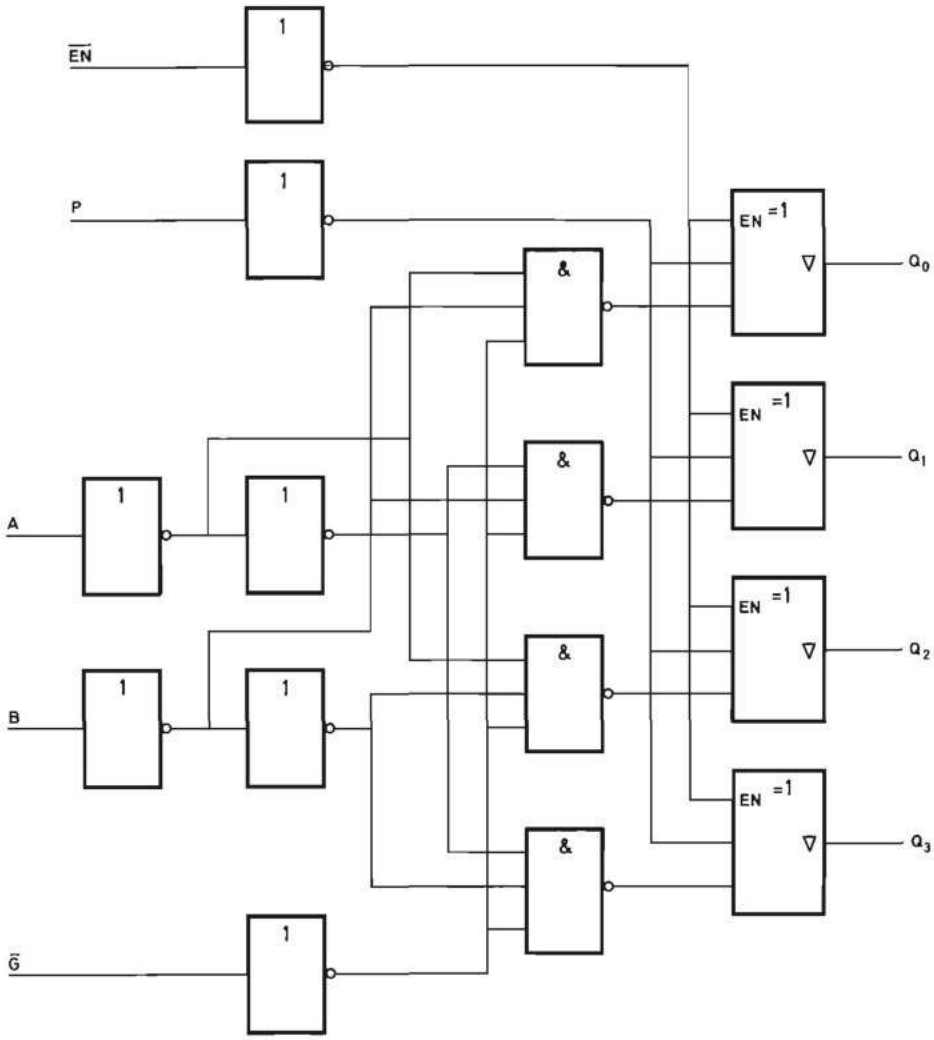


FIGURA 3.60.—**Esquema de un decodificador de uno entre cuatro con variable de desinhibición, de inversión y de control del tercer estado de las salidas.**

— Una variable de control de inversión o de polaridad de salida P que según se encuentre en nivel cero o uno hace que las salidas sean activas con uno o cero respectivamente.
— Una variable $\overline{EN}$ de control del tercer estado de las salidas que hace que todas se pongan en dicho estado cuando ella se encuentra en nivel uno.

En la tabla 3.20 se representa la tabla de verdad del decodificador de la figura 3.60. Se recomienda al lector que la analice para comprobar la veracidad de lo indicado en los párrafos anteriores. En la figura 3.61 se representa el símbolo nor-

$\overline{EN}$	P	$\overline{G}$	B	A	Q_0	Q_1	Q_2	Q_3
1	X	X	X	X	Tercer Estado			
0	0	1	X	X	0	0	0	0
0	1	1	X	X	1	1	1	1
0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	1	0	0
0	0	0	1	0	0	0	1	0
0	0	0	1	1	0	0	0	1
0	1	0	0	0	0	1	1	1
0	1	0	0	1	1	0	1	1
0	1	0	1	0	1	1	0	1
0	1	0	1	1	1	1	1	0

TABLA 3.20.—**Tabla de verdad del decodificador uno entre cuatro de la figura 3.60.**

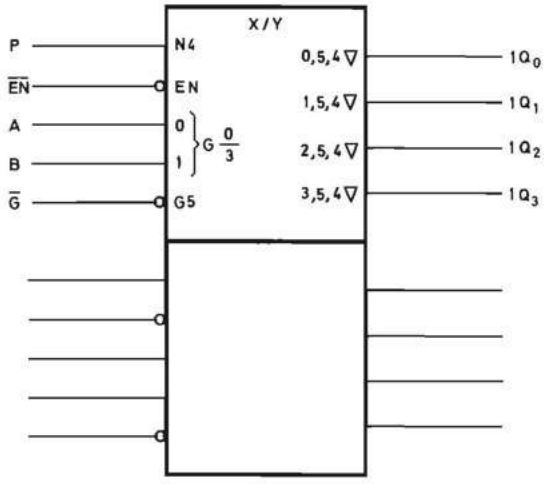


FIGURA 3.61.—**Símbolo lógico normalizado de un doble decodificador como el de la figura 3.60.**

malizado de un doble decodificador como el descrito que constituye un ejemplo de la combinación de simultaneidad de relaciones de dependencia descrita en el apartado A1.4.3.8 del apéndice 1. Al lector que no conozca la simbología normalizada se le recomienda nuevamente la lectura del citado apéncice.

Una aplicación muy interesante de los decodificadores no excitadores en combinación con una puerta NO-Y es la generación de funciones lógicas (fig. 3.62). La función se genera conectando a las entradas de la puerta NO-Y las salidas correspondientes a los productos canónicos que toman el valor uno.

Mediante el decodificador decimal se puede generar cualquier función de tres variables conectando éstas a laș entradsa 1, 2 y 4. Las salidas 0 a 7 toman el valor

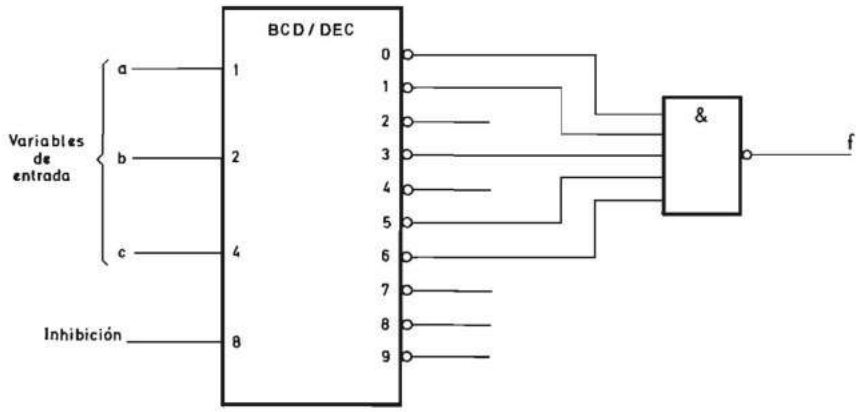


FIGURA 3.62.—**Generación de funciones lógicas mediante un decodificador decimal.**

cero cuando en aquéllas se presenta la combinación correspondiente y, por tanto, cada una de ellas realiza un producto canónico de las variables de entrada. La entrada 8 se puede utilizar como inhibidora, manteniéndola en estado lógico cero y haciéndola subir al estado uno cuando se desea inhibir la generación de la función. En la figura 3.62 se representa el circuito que genera la función de tres variables $f = \sum_{3}(0, 1, 3, 5, 6)$.

En la figura 3.63 se indica la utilización del decodificador hexadecimal como generador de funciones de cuatro variables a, b, c y d en combinación con una puerta NO-Y.

En dicha figura se generan las funciones:

$$f_1\ (a, b, c, d) = \sum_{4}\ (0, 9, 15)$$

$$f_2\ (a, b, c, d) = \sum_{4}\ (1, 2, 5, 10)$$

De igual forma con el decodificador uno entre cuatro se puede generar cualquier función de dos variables.

Los decodificadores excitadores tal como se indicó al principio de este apartado se diseñan para activar cargas externas como por ejemplo indicadores numéricos, relés electromagnéticos u optoacopladores. Sus salidas admiten unas tensiones mayores y poseen una cargabilidad más elevada que los demás bloques funcionales combinacionales de la misma tecnología.

En el diseño de un decodificador excitador surgen diversas variantes que hacen que existan diferentes circuitos en la práctica.

En la figura 3.64 se representa el símbolo normalizado de un decodificador excitador de decimal codificado en binario natural (BCD) a decimal en el que las salidas son de colector abierto y se indican con el símbolo correspondiente (ver apartado A1.3 del apéndice 1). Debajo del indicativo se coloca el símbolo ▷ que especifica que el circuito posee amplificadores (drivers) a la salida que le dan las características eléctricas anteriormente indicadas.

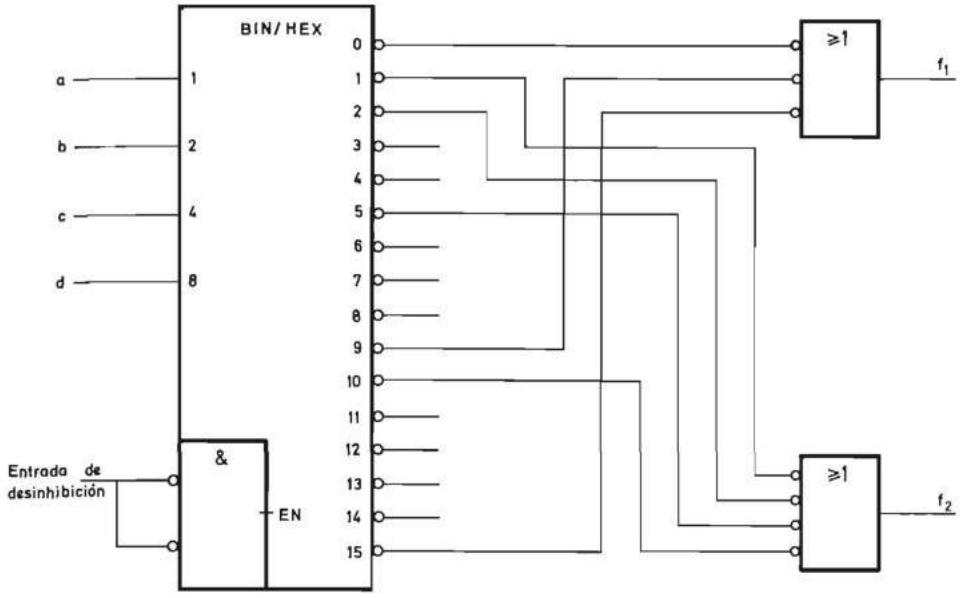


FIGURA 3.63.—**Generación de funciones lógicas con un decodificador hexadecimal.**

Aunque no es muy correcto, los fabricantes de circuitos integrados suelen llamar decodificadores excitadores de BCD a 7 segmentos (BCD to seven segment decoder/driver) a los convertidores de código que cambian la información de un código binario a otro distinto del decimal y que además poseen amplificadores a la salida. Uno de los más utilizados es el que convierte del BCD natural al de 7 segmentos de los visualizadores (displays) de estado sólido.

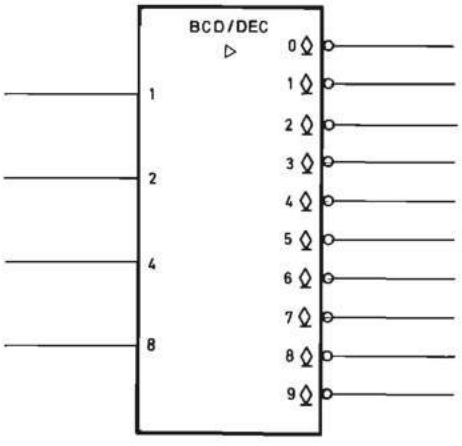


FIGURA 3.64.—**Símbolo lógico de un decodificador excitador (decoder driver) decimal.**

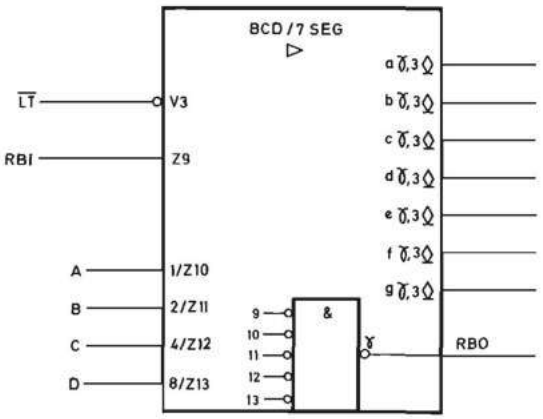


FIGURA 3.65.—**Símbolo lógico de un decodificador excitador de BCD a siete segmentos.**

En la figura 3.65 se representa el símbolo lógico de un decodificador excitador de decimal codificado en binario natural (BCD) a un código de 7 segmentos adecuado para gobernar visualizadores numéricos de diodos emisores de luz con el citado número de segmentos o barras. Las salidas del circuito son de colector abierto y por ello en ellas se coloca el símbolo ◇. La tabla de verdad del circuito se representa en la tabla 3.21. Las salidas reciben las denominaciones «*a*, *b*, *c*, *d*, *e*, *f*, *g*», que corresponden a las barras de un visualizador de 7 segmentos representado en la figura 3.66*b*.

Los estados de las salidas reciben la denominación «*S*» (correspondiente a transistor saturado) o «*C*» (correspondiente a transistor cortado). El circuito posee una entrada de prueba $\overline{LT}$ (Lamp Test) activa con un cero lógico que cuando se en-

$\overline{LT}$	*RBI*	*D*	*C*	*B*	*A*	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*RBO*
1	1	0	0	0	0	S	S	S	S	S	S	C	1
1	X	0	0	0	1	C	S	S	C	C	C	C	1
1	X	0	0	1	0	S	S	C	S	S	C	S	1
1	X	0	0	1	1	S	S	S	S	C	C	S	1
1	X	0	1	0	0	C	S	S	C	C	S	S	1
1	X	0	1	0	1	S	C	S	S	C	S	S	1
1	X	0	1	1	0	C	C	S	S	S	S	S	1
1	X	0	1	1	1	S	S	S	C	C	C	C	1
1	X	1	0	0	0	S	S	S	S	S	S	S	1
1	X	1	0	0	1	S	S	S	C	C	S	S	1
1	0	0	0	0	0	C	C	C	C	C	C	C	0
0	X	X	X	X	X	S	S	S	S	S	S	S	1

TABLA 3.21

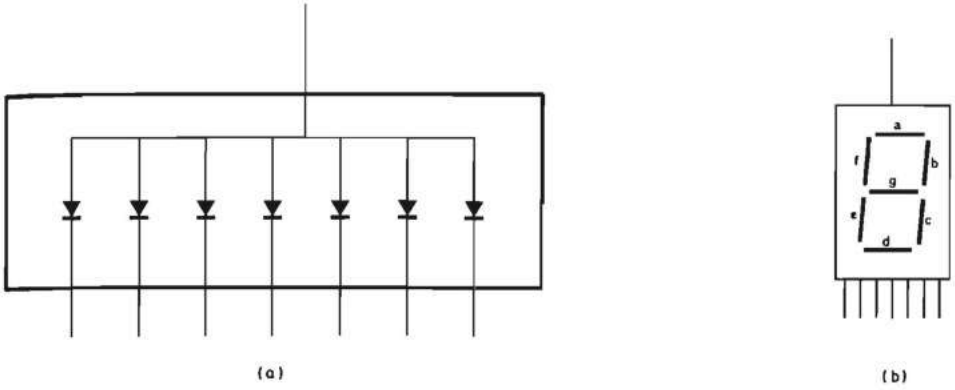


FIGURA 3.66.—**Visualizazor de siete segmentos: *a*) montaje de diodos emisores de luz (LED) con ánodo común; *b*) realización física.**

cuentra en este estado satura todos los transistores y por ello se indica con el símbolo V (correspondiente a la relación de dependencia 0 estudiada en el apartado A1.4.3.2 del apéndice 1). Además tiene una entrada de inhibición o de propagación de cero *RBI* (Ripple Blanking Input) que cuando está en nivel cero inhibe la visualización si el número a visualizar es el cero en binario. La entrada *RBI* está ligada con la salida de propagación de ceros [*RBO* (Ripple Blanking Output)], que está siempre en uno excepto en la situación que se acaba de describir. Para ello se genera la variable γ (figura 3.65) que constituye una función NO-Y del inverso de las entradas *A*, *B*, *C* y *D* y el inverso de *RBI*. Cada salida que se aplica al visualizador incorpora una letra de la «*a*» a la «*g*» [correspondiente al segmento del visualizador sobre el que actúa (figura 3.66 y tabla 3.21)] seguida de la letra γ. Ello significa que para que esta salida se active [transistor saturado (S)] ha de aplicarse a las entradas *A*, *B*, *C* y *D* una combinación binaria para la cual se deba encender la barra correspondiente y simultáneamente ha de estar γ en nivel uno. Por ello la relación entre el terminal γ y los terminales de salida de los segmentos es la relación *Y*.

Además todas las salidas que se aplican al visualizador deben activarse si se pone a cero el terminal $\overline{LT}$. Por ello cada terminal incorpora el número 3. La «,» entre γ y 3 indica que γ y *V*3 actúan simultáneamente.

Este decodificador permite visualizar números representados en BCD natural mediante visualizadores realizados con diodos emisores de luz conectados en un montaje de ánodo común (figura 3.66*a*), realizando el circuito de la figura 3.67 en el que las resistencias *R* limitan la corriente a través de los diodos. La utilidad de la entrada *RBI* y la salida *RBO* se comprueba con el montaje de la figura 3.68 que representa un visualizador de tres dígitos, en el que no se visualizan los ceros no significativos.

En efecto, la conexión en serie (ripple) de las entradas *RBI* y salidas *RBO* hace que un determinado dígito solamente se active si el número presente a la entrada de su visualizador es distinto de cero o bien si siendo igual a cero, es distinto de cero alguno de los que le preceden.

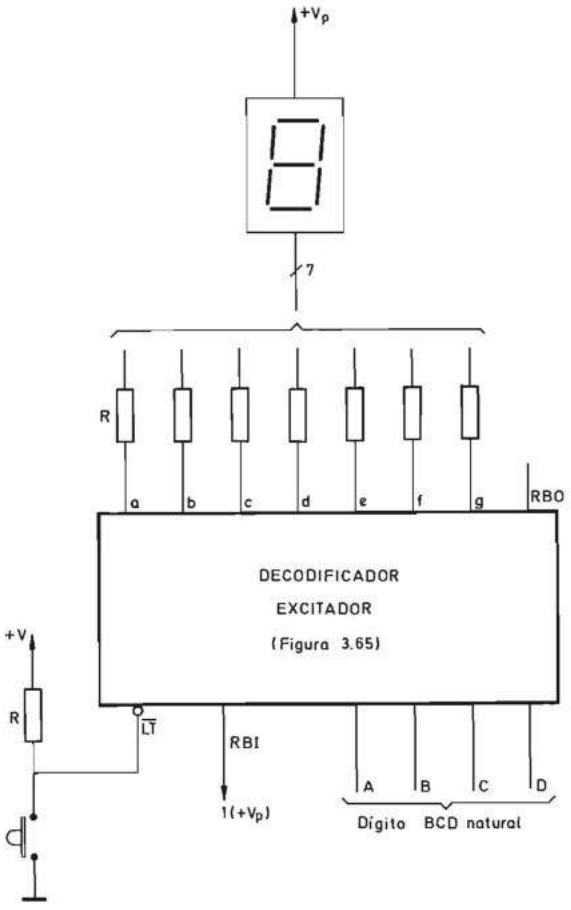


FIGURA 3.67.—**Esquema de un visualizador de un digito BCD natural realizado con el visualizador de la figura 3.66.**

b) Demultiplexores.

El circuito decodificador puede ser utilizado como demultiplexor. Un circuito demultiplexor es un sistema combinacional con una entrada de información D y m salidas, que posee además n entradas de selección tal que $2^n \geq m$. La información D se puede hacer aparecer en cualquiera de las salidas aplicando a las entradas de selección la combinación binaria adecuada.

En la figura 3.69 se representa una aplicación del decodificador decimal como demultiplexor de ocho canales. La combinación binaria presente en las entradas 0, 1 y 2 (equivalentes a 1, 2 y 4 respectivamente de la figura 3.52 *b*) hace que la información aplicada en la entrada 3 (equivalente a 8 en la figura 3.52*b*) aparezca en una de las salidas 0 a 7. Dicha combinación realiza, por lo tanto, la función *Y (G)* con las salidas 0 a 7 y, por ello, se le asigna el indicativo *G* 0/7. Por otra parte, las salidas 8 y 9 son el in-

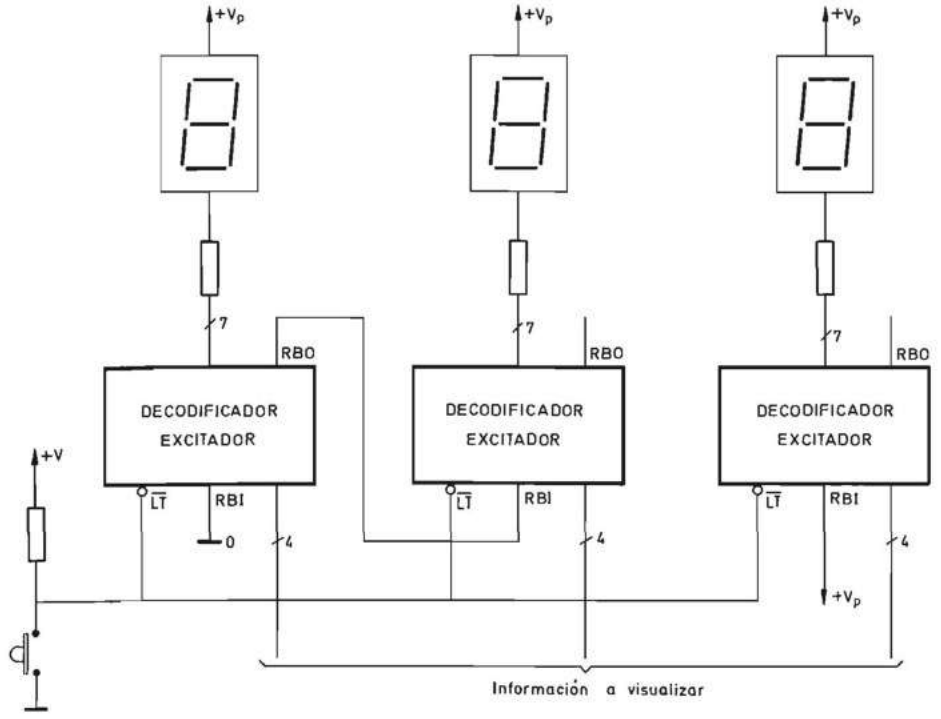


FIGURA 3.68.—**Esquema de tres visualizadores de siete segmentos con inhibición de la visualización de los ceros no significativos.**

verso de la 0 y de la 1 respectivamente. En la figura 3.69 se ha cambiado también la denominación (DMUX en lugar de BCD/DEC) del símbolo y la de los terminales de salida para adaptarlos a la nueva función del circuito.

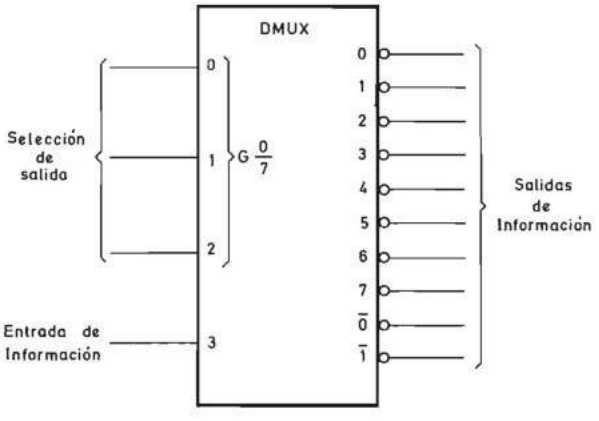


FIGURA 3.69.—**Decodificador decimal utilizado como demultiplexor.**

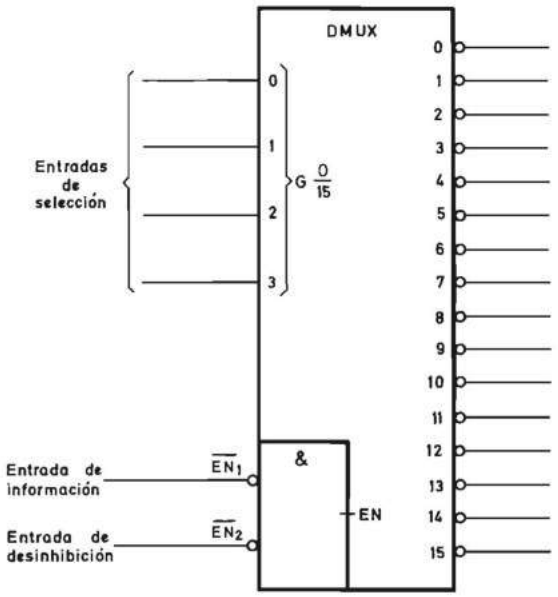


FIGURA 3.70.—**Decodificador hexadecimal utilizado como demultiplexor.**

El decodificador hexadecimal puede también ser utilizado como demultiplexor. En la figura 3.70 se representa el símbolo normalizado. Las entradas 1, 2, 4 y 8 de la figura 3.54 pasan a llamarse 0, 1, 2 y 3 respectivamente, y se les asignan los símbolos *G* 0/15 que indican que se realiza la función *Y (G)* de cada combinación con la entrada de información para obtener las salidas 0 a 15. Una de las entradas de desinhibición (en la figura 3.70 la $\overline{EN}_1$) es la entrada de información y la otra (en la figura 3.70 la $\overline{EN}_2$) es la de desinhibición propiamente dicha.

En la figura 3.71 se presenta una aplicación del demultiplexor de la figura 3.70. Las salidas del multiplexor se conectan a sendos contadores C0 a C15 que reciben los impulsos de una entrada común a todos ellos. Cada uno posee una entrada de inhibición G1 que según se encuentre en cero o uno permite o no que se realice el contaje de los impulsos. Cada entrada de inhibición se conecta a una salida del demultiplexor. Las entradas de selección *G* 0/15 de éste, seleccionan el canal de salida 0 a 15 en el que aparece el estado de la entrada de información. De esta forma se logra que el contador correspondiente realice o no el contaje de los impulsos según la entrada de información $\overline{EN}_1$ se encuentre en nivel cero o uno respectivamente (siempre y cuando $\overline{EN}_2$ se encuentre en nivel cero).

3.7.2 Codificadores

Los circuitos codificadores son sistemas combinacionales de 2^n entradas y n salidas realizados de tal forma que, cuando una sola de las entradas adopta un estado lógico determinado cero o uno, a la salida aparece la combinación binaria

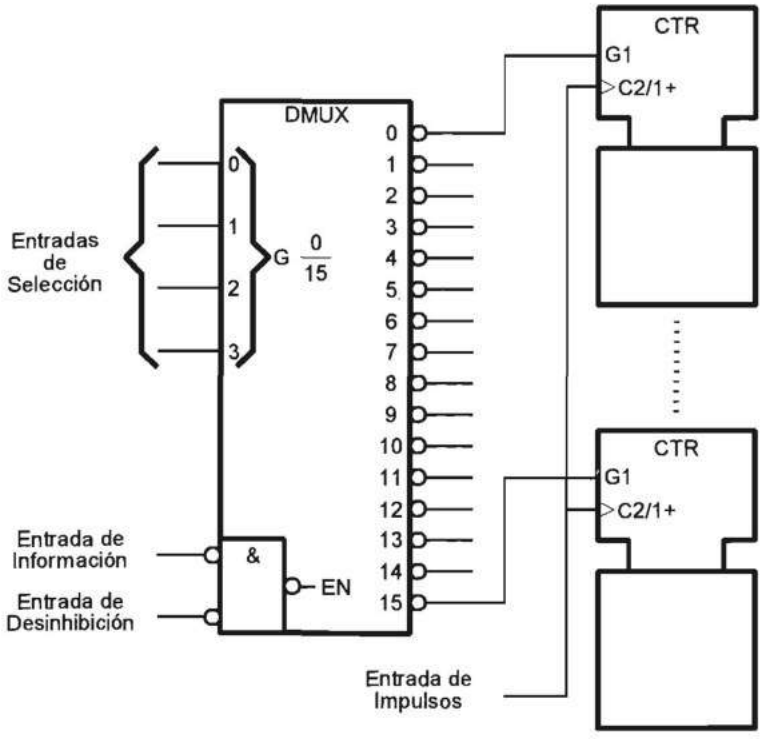


FIGURA 3.71.—**Ejemplo de aplicación del decodificador hexadecimal como demultiplexor.**

correspondiente al número decimal asignado a dicha entrada. Realizan por tanto la función inversa de los decodificadores.

Los circuitos codificadores pueden ser diseñados con prioridad o sin ella. En los codificadores sin prioridad, cuando más de una entrada toma el estado activo uno o cero, la combinación de salida posee tantos unos como hay en las correspondientes a la excitación de cada una de las entradas independientemente; por lo que en este codificador solamente debe de ser activa una entrada en cada instante.

Los codificadores con prioridad codifican la entrada activa de mayor valor decimal sin tener en cuenta las demás. Su tabla de verdad se representa en la tabla 3.22 para el caso en que $n = 3$ y las entradas son activas con un cero lógico.

En la tabla 3.22 se observa que, debido a que las tres variables binarias Q_2, Q_1 y Q_0 solamente tienen ocho combinaciones diferentes, no es posible discernir entre la situación en que ninguna de las entradas está activada y aquella en que es activa la de mayor peso. Para lograrlo se puede añadir al circuito una nueva salida que detecte alguna de ellas. Además, se puede disponer una entrada de inhibición que fuerce a todas las salidas a un estado determinado, independientemente del estado de las restantes entradas.

La tabla 3.23 representa la tabla de verdad de un codificador con prioridad de $n = 3$ que posee todas las salidas y entradas descritas. Cuando la entrada I de

A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Q_2	Q_1	Q_0
1	1	1	1	1	1	1	1	1	1	1
X	X	X	X	X	X	X	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0
X	X	X	X	X	0	1	1	1	0	1
X	X	X	X	0	1	1	1	1	0	0
X	X	X	0	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	0	1	0
X	0	1	1	1	1	1	1	0	0	1
0	1	1	1	1	1	1	1	0	0	0

TABLA 3.22

I	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Q_2	Q_1	Q_0	P_0	P_1
1	X	X	X	X	X	X	X	X	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	X	X	X	X	X	X	X	0	0	0	0	0	1
0	X	X	X	X	X	X	0	1	0	0	1	0	1
0	X	X	X	X	X	0	1	1	0	1	0	0	1
0	X	X	X	X	0	1	1	1	0	1	1	0	1
0	X	X	X	0	1	1	1	1	1	0	0	0	1
0	X	X	0	1	1	1	1	1	1	0	1	0	1
0	X	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

X Las entradas correspondientes pueden tomar los valores 0 o 1 lógicos.

TABLA 3.23

inhibición toma el valor lógico uno, todas las salidas adoptan dicho estado. Cuando I es cero y todas las demás entradas están en uno, sólo P_1 adopta el estado cero.

Cuando, estando I en cero, cualquiera de las restantes entradas se encuentra en dicho estado, la salida P_0 adopta el estado cero y en las salidas Q_2, Q_1 y Q_0 aparece la combinación binaria equivalente al complemento a siete de la entrada de mayor peso que se encuentra en estado cero.

En la figura 3.72 se representa el símbolo lógico de este circuito en sus versiones no normalizada (*a*) y normalizada (*b*). Este circuito constituye un ejemplo de cómo los símbolos normalizados permiten prescindir de las tablas de verdad. En primer lugar, el indicativo HPRI/BIN indica que se trata de un codificador con

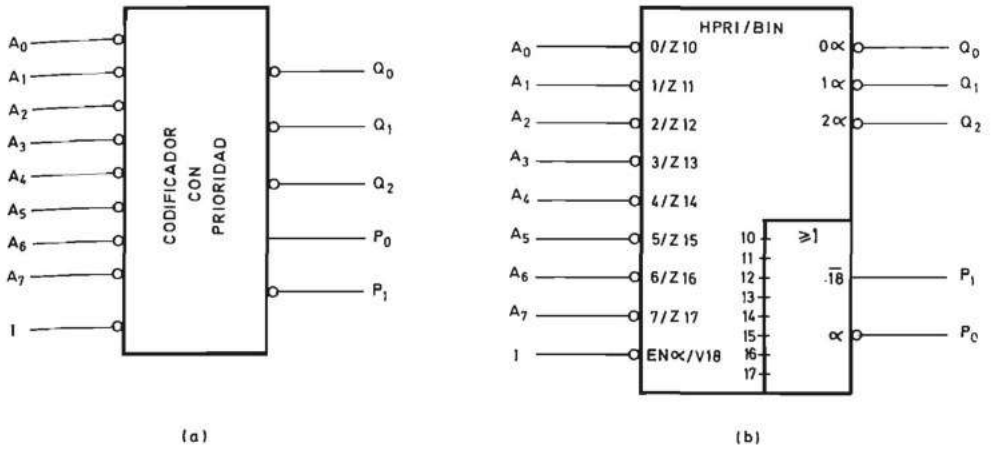


FIGURA 3.72.—**Símbolos lógicos de un codificador con prioridad: *a*) no normalizado; *b*) normalizado.**

prioridad con salida en binario natural. Las salidas codificadas Q_0, Q_1 y Q_2 reciben internamente las denominaciones 0α, 1α y 2α. Los números 0, 1 y 2 indican el peso de cada bit y el símbolo α establece que la salida correspondiente se pone a cero si ENα está en nivel cero. (El terminal externo se pone a uno debido a la existencia de una inversión).

Las entradas que se codifican se identifican con los números 0 a 7 correspondientes a su peso en decimal. Además, cada una de ellas posee el símbolo Z seguido de un número que indica que está unida a la correspondiente entrada de la puerta O que genera P_1 y P_0; por ejemplo, la entrada Z14 del circuito se une a la entrada 14 de la puerta O.

La entrada de inhibición se denomina ENα/V18 y posee un símbolo de inversión. Al mismo tiempo P_1 lleva el símbolo $\overline{18}$ y P_0 el símbolo α y está seguido de una inversión. Todo ello indica que P_1 y P_0 responden a las ecuaciones:

$$P_1 = \overline{A}_0 + \overline{A}_1 + \overline{A}_2 + \overline{A}_3 + \overline{A}_4 + \overline{A}_5 + \overline{A}_6 + \overline{A}_7 + I$$

$$P_0 = (\overline{A}_0 + \overline{A}_1 + \overline{A}_2 + \overline{A}_3 + \overline{A}_4 + \overline{A}_5 + \overline{A}_6 + \overline{A}_7)\,\overline{I}$$

La salida P_0 ($\overline{18}$) permite la realización con gran facilidad de un codificador de mayor número de entradas, porque adopta el estado cero cuando cualquiera de las entradas A_0 a A_7 es activa [ver Manual de prácticas de electrónica digital (diseño 4.6) de Editorial Marcombo].

Se sugiere al lector que realice el diseño de ese circuito con puertas como práctica.

En la figura 3.73 se representa el símbolo lógico normalizado de un codificador con prioridad de decimal a decimal codificado en binario natural (BCD natural). Se sugiere al lector que a partir de este símbolo deduzca la tabla de verdad de la tabla 3.24.

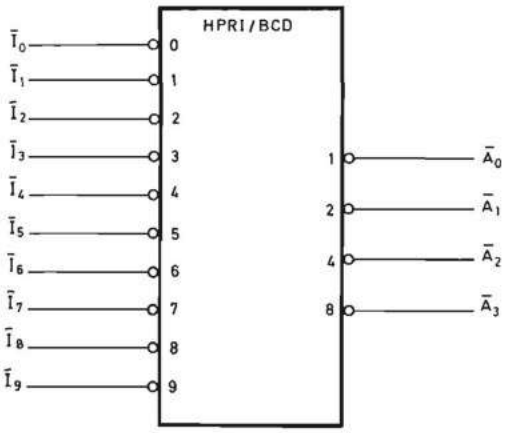


FIGURA 3.73.—**Símbolo lógico normalizado de un codificador con prioridad de decimal a decimal codificado en binario natural (BCD natural).**

$\overline{I_0}$	$\overline{I_1}$	$\overline{I_2}$	$\overline{I_3}$	$\overline{I_4}$	$\overline{I_5}$	$\overline{I_6}$	$\overline{I_7}$	$\overline{I_8}$	$\overline{I_9}$	$\overline{A_3}$	$\overline{A_2}$	$\overline{A_1}$	$\overline{A_0}$
1	1	1	1	1	1	1	1	1	1	1	1	1	1
X	X	X	X	X	X	X	X	X	0	0	1	1	0
X	X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	1	0	1	1	1	1	1	0	1	0
X	X	X	1	0	1	1	1	1	1	1	0	1	1
X	X	X	0	1	1	1	1	1	1	1	1	0	0
X	X	0	1	1	1	1	1	1	1	1	1	0	1
X	0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1

TABLA 3.24.—**Tabla de verdad de un codificador decimal con prioridad.**

3.7.3 Multiplexores

Los multiplexores son circuitos combinacionales que poseen n canales de entrada, uno de salida y m entradas de selección, siendo $n = 2^m$ que permiten elegir cuál es el canal de entrada cuya información aparece en el de salida.

Existen diversas formas de realizar los multiplexores, que se estudian en apartados sucesivos.

3.7.3.1 Multiplexores realizados con puertas Y y puertas O. Una de las formas de realizar un multiplexor es mediante un número n de puertas Y conectadas a una única puerta O, que poseen m entradas de selección E (tal que $2^m = n$). Cada combinación binaria presente en las entradas de selección produce la aper-

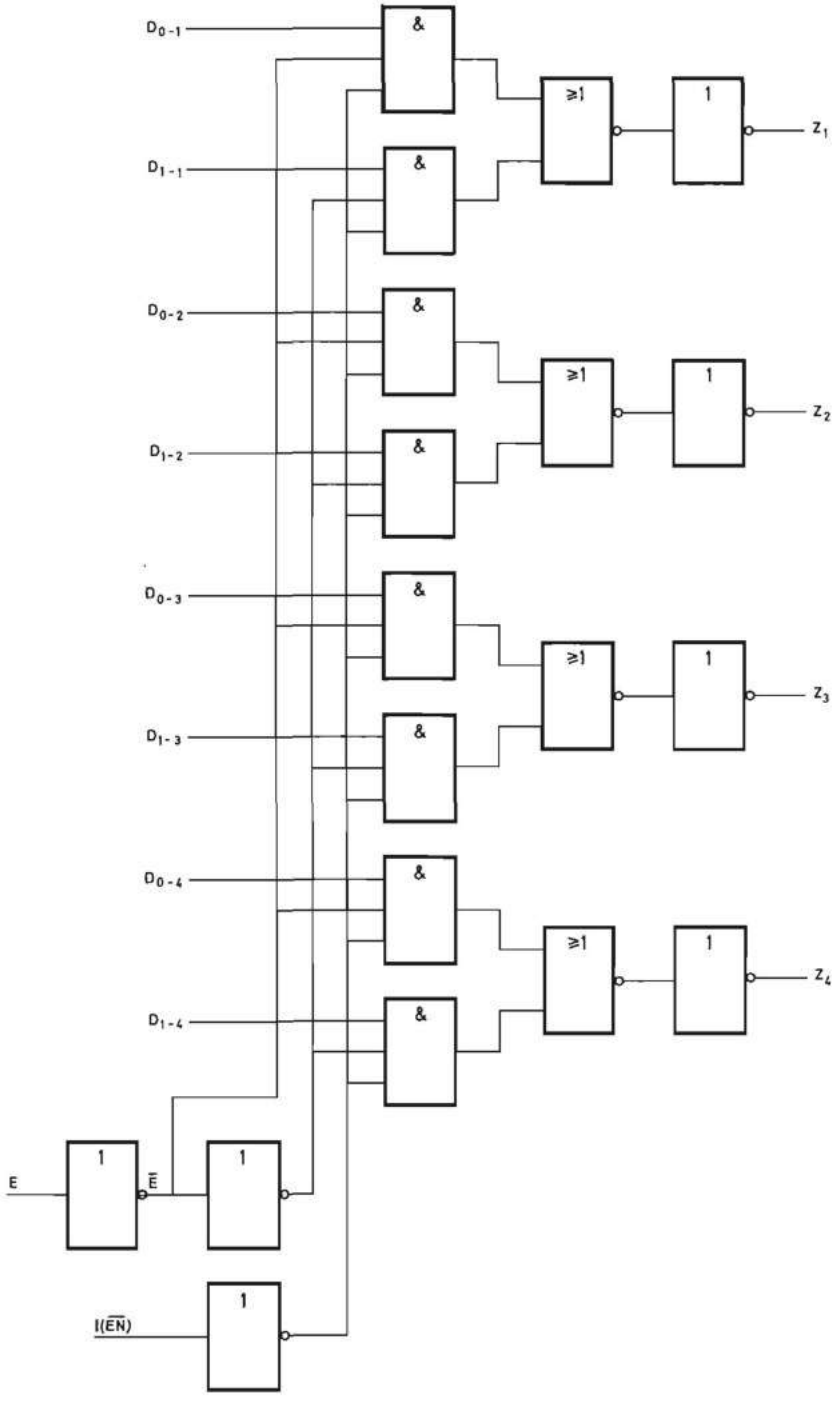


FIGURA 3.74.—**Esquema de un cuádruple multiplexor de dos canales.**

tura de una puerta Y únicamente, de tal forma que la información presente en la entrada D correspondiente puede pasar a la salida Z de la puerta O. Por ello, las entradas de selección reciben en la nueva simbología la denominación $G\frac{0}{2^m - 1}$. Eligiendo adecuadamente la combinación binaria presente en ellas, se logra que en la salida Z aparezca la información presente en cualquiera de las entradas D y el circuito actúa entonces como un conmutador electrónico.

Existen circuitos integrados que contienen cuatro multiplexores de dos canales, dos multiplexores de cuatro canales, un multiplexor de ocho canales y un multiplexor de dieciséis canales.

En la figura 3.74 se representa el esquema lógico de un cuádruple multiplexor de dos canales que posee una entrada de selección E ($G1$) común a todos ellos. Este circuito posee además una entrada de inhibición/desinhibición I ($\overline{EN}$) que cuando se encuentra en nivel uno pone a cero todas las salidas de las puertas Y y fuerza las salidas Z al estado lógico cero, independientemente de la información aplicada en la entrada de selección E y en las entradas de información D. En la figura 3.75 se representa el símbolo lógico no normalizado (a) y normalizado (b) de este multiplexor. Las entradas EN y $G1$ del símbolo normalizado se colocan en la parte superior porque son comunes a los cuatro multiplexores. En los canales de entrada se colocan los símbolos $\overline{1}$ y 1 que indican que la información del primero aparece en la salida si $G1$ está en nivel cero y por el contrario es la del segundo la que aparece si $G1$ está en nivel uno.

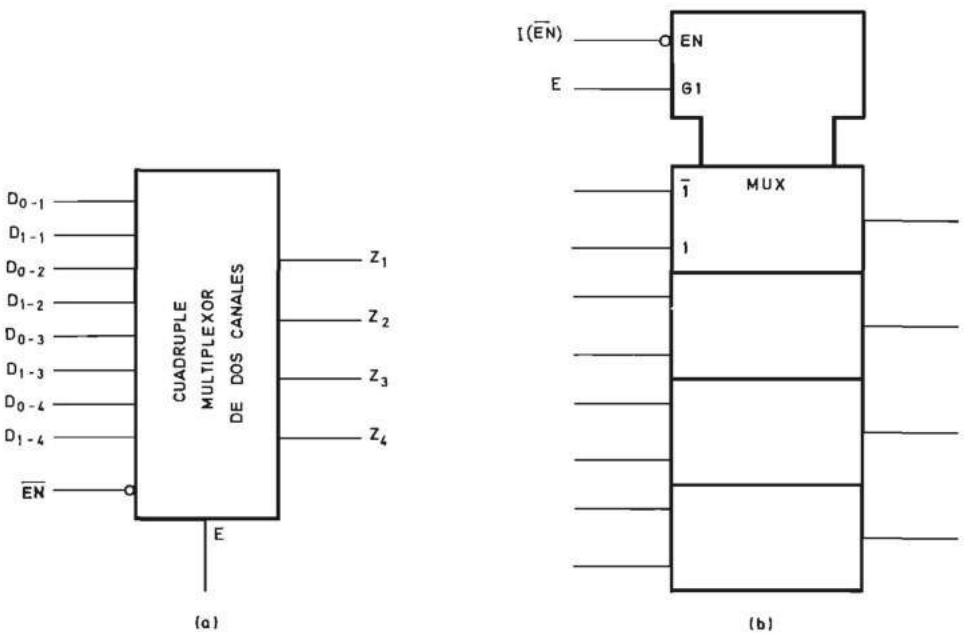


FIGURA 3.75.—**Símbolos lógicos del cuádruple multiplexor de dos canales de la figura 3.74: *a*) no normalizado; *b*) normalizado.**

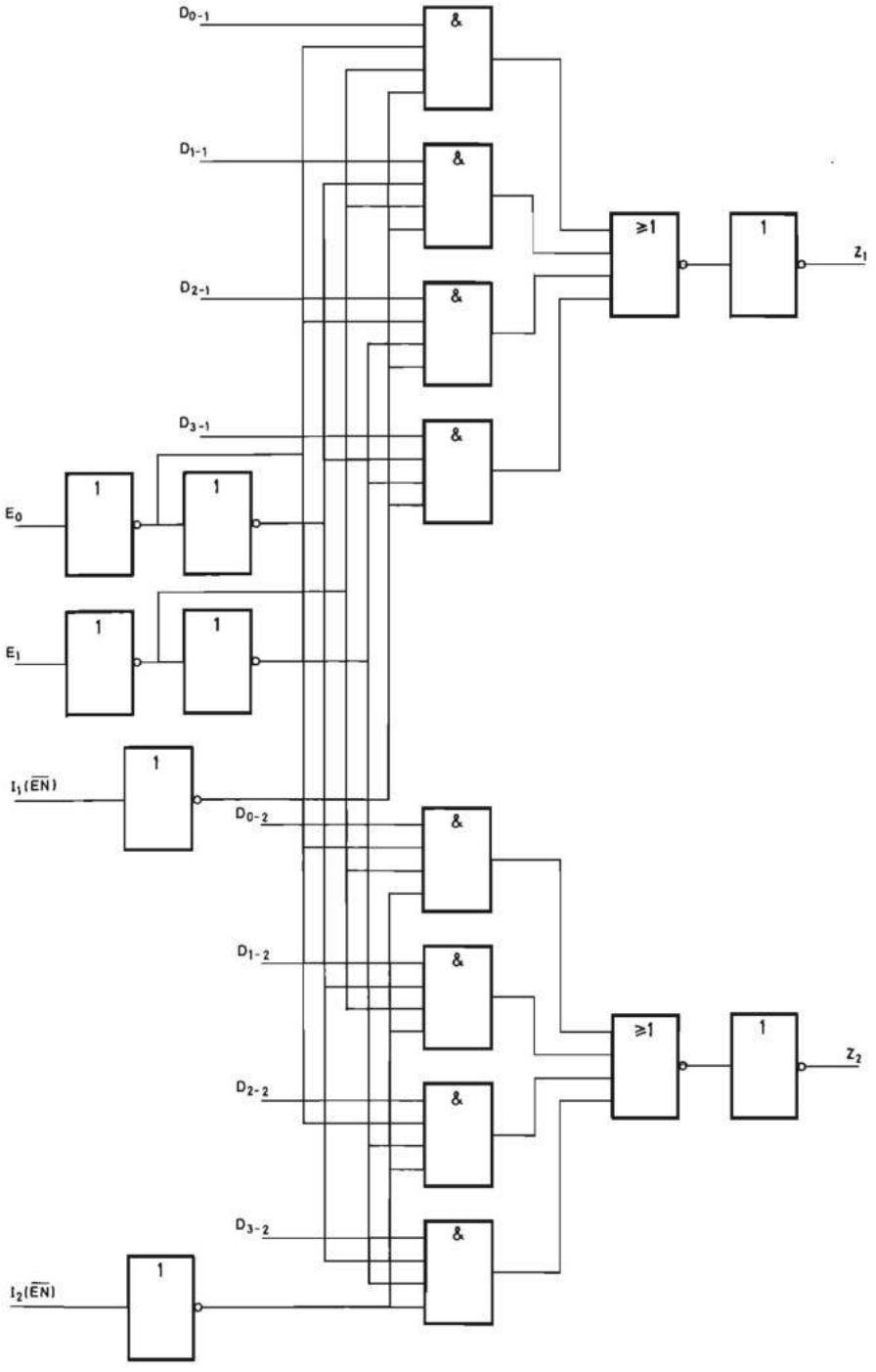


FIGURA 3.76.—**Esquema lógico de un doble multiplexor de cuatro canales.**

En la figura 3.76 se representa el esquema lógico de un doble multiplexor de cuatro canales con entradas de selección E_0 y E_1 (G 0/3) comunes a todos y entradas de inhibición I_1 e I_2 ($\overline{EN}$) independientes. En la figura 3.77 se representan los símbolos lógicos no normalizado y normalizado. En este último, en la parte superior se colocan solamente las entradas de selección E_1 (G 0/3) porque cada multiplexor posee una entrada de desinhibición (EN) independiente.

En la figura 3.78 se representa el esquema de un multiplexor de ocho canales con tres variables de selección E_0, E_1 y E_2 (G 0/7) y una variable de inhibición/desinhibición I ($\overline{EN}$), que si se encuentra en nivel uno pone a nivel cero la salida. En la figura 3.79 se representan los símbolos lógicos no normalizado (a) y normalizado (b) correspondientes a este multiplexor. En el caso de que la entrada EN ponga en tercer estado las salidas, se coloca el símbolo ▽ en éstas (figura 3.80).

La figura 3.81 representa un multiplexor de dieciséis canales con cuatro variables de selección E_0, E_1, E_2 y E_3 (G 0/15) que posee una entrada de inhibición I ($\overline{EN}$) que si está en estado uno, pone a cero la salida. En la figura 3.82 se representan los símbolos lógicos correspondientes.

En la figura 3.83 se representa el símbolo de un séxtuple multiplexor de dos canales de entrada A y B que puede presentar en su salida uno u otro o el producto de ambos en forma directa o inversa. Este circuito constituye un ejemplo claro de las ventajas de utilización de los símbolos normalizados. Las variables de selección de entrada E_0 y E_1 pueden encontrarse en los cuatro estados posibles, 0, 1, 2 y 3.

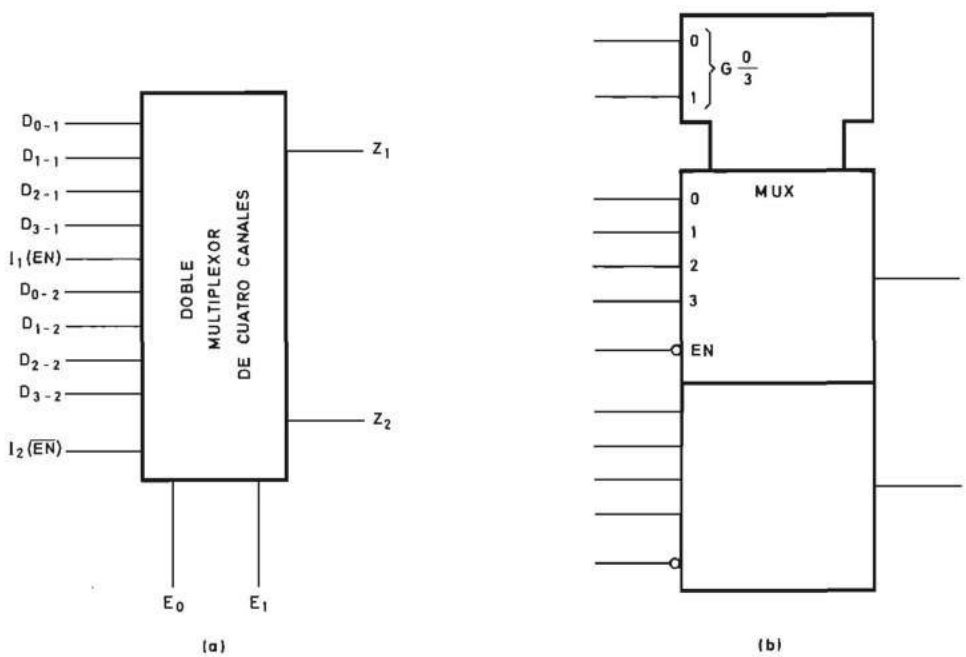


FIGURA 3.77.—**Símbolos lógicos del doble multiplexor de cuatro canales de la figura 3.76: *a*) no normalizado; *b*) normalizado.**

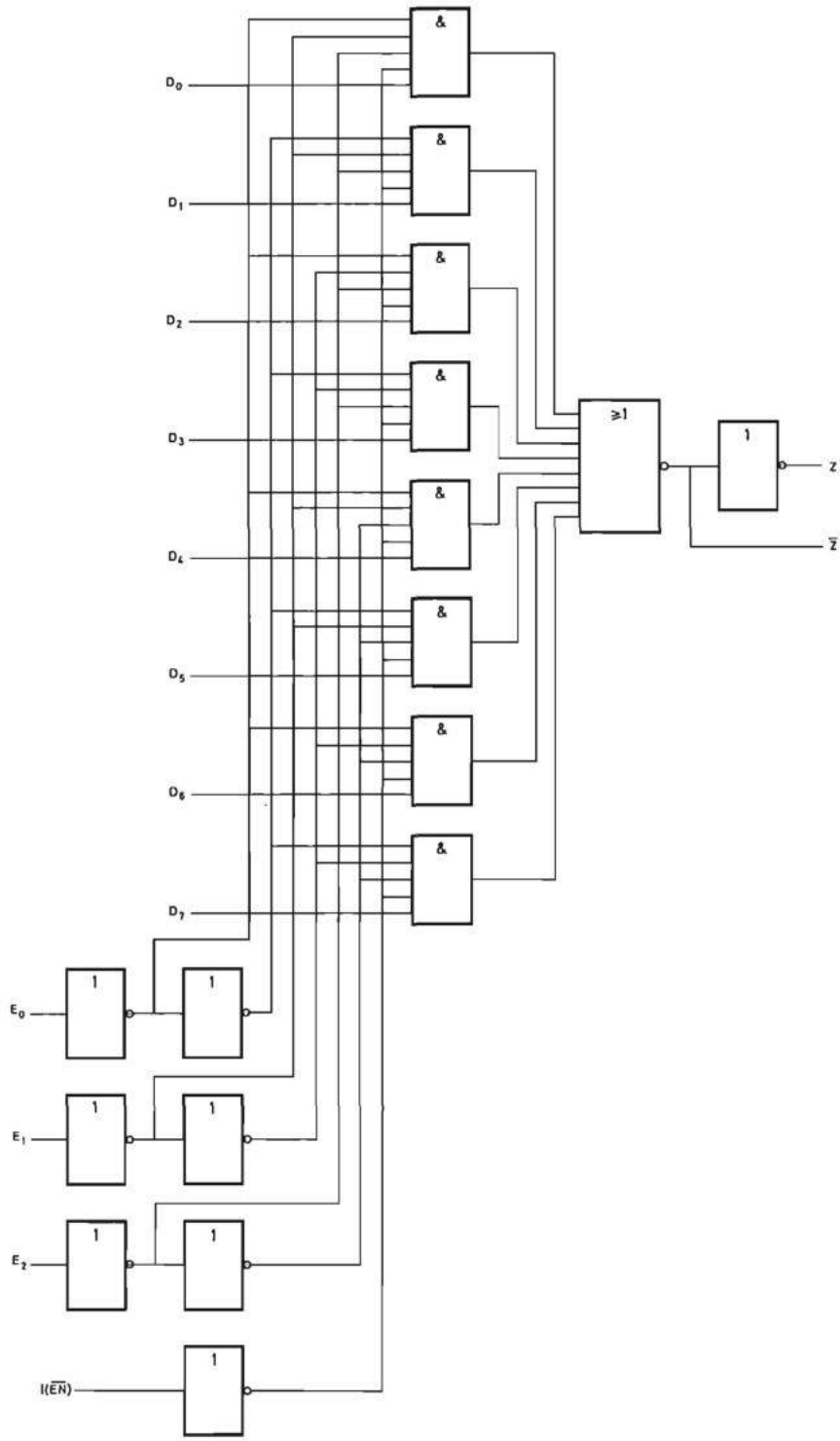


FIGURA 3.78.—**Esquema lógico de un multiplexor de ocho canales.**

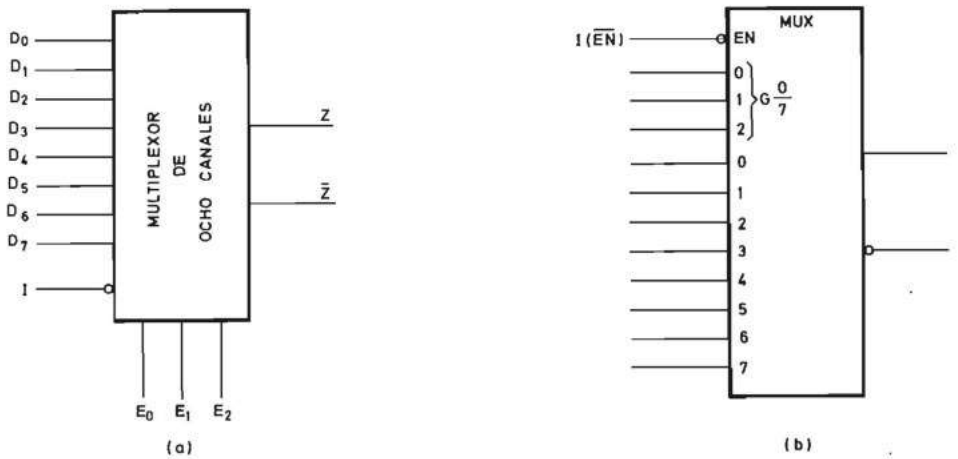


FIGURA 3.79.—**Símbolos lógicos del multiplexor de ocho canales de la figura 3.78: *a*) no normalizado; *b*) normalizado.**

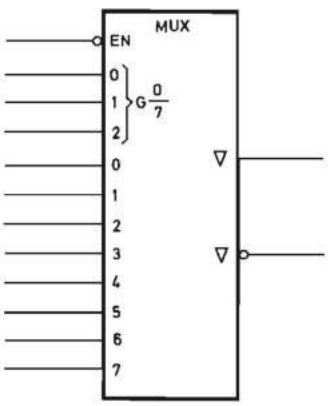


FIGURA 3.80.—**Símbolo lógico normalizado de un multiplexor de ocho canales con salidas de tres estados.**

Estas variables actúan mediante la relación de dependencia Y (G) con las entradas del multiplexor A y B y su producto. La asignación del 0 a la entrada A indica que este canal se selecciona cuando E_0 y E_1 se encuentran ambas en estado cero. La combinación E_1, E_0 = 11 (3) no selecciona ningún canal de entrada y hace que la salida se ponga a nivel cero.

La salida está afectada además simultáneamente por la entrada $N4$ y la variable interna $EN5$. Cuando la variable $N4$ se encuentra en nivel uno la salida se invierte siempre y cuando E_0 y E_1 se encuentren en 00, 01, 10.

La variable $EN5$ es igual a: $EN5 = \overline{COMP \cdot E_1 \cdot E_0}$. Por lo tanto, si E_1 E_0 = 11 (3) y $COMP$ ($N4$) se encuentra en nivel uno, resulta $EN5$ = 0 y la salida se pone en tercer estado.

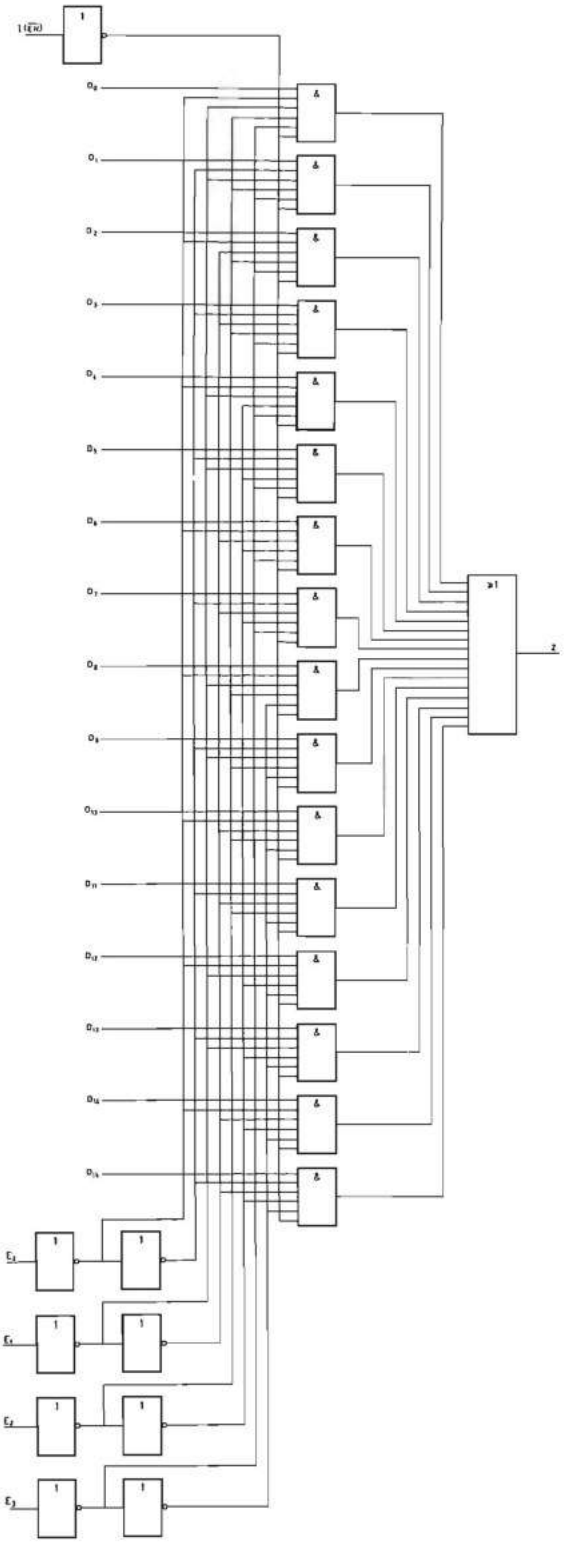


FIGURA 3.81.—**Esquema lógico de un multiplexor de dieciséis canales.**

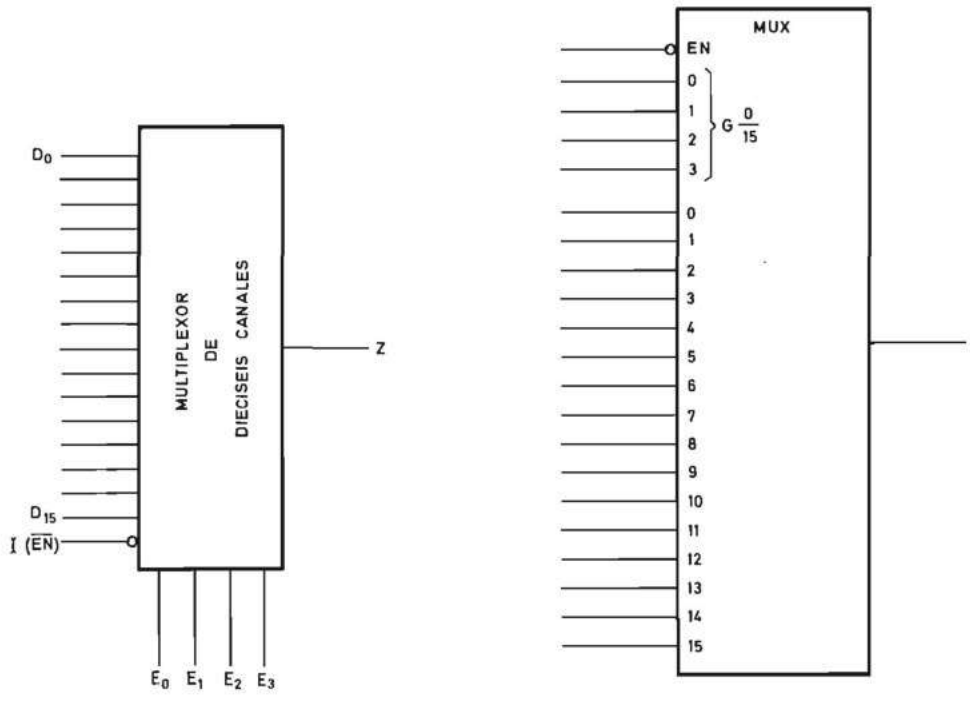


FIGURA 3.82.—**Símbolos lógicos del multiplexor de dieciséis canales de la figura 3.81: *a*) no normalizado; *b*) normalizado.**

El lector puede deducir la tabla de verdad de este multiplexor, que es la indicada en la tabla 3.25 y comprobar que la información contenida en ella se deduce directamente del símbolo normalizado de la figura 3.83, lo que la hace superflua.

Como ejemplo, en la tabla 3.26 se representa también la tabla de verdad del multiplexor de ocho canales de entrada de las figuras 3.78 y 3.79. El lector puede com-

COMP	E_1	E_0	Salida
0	0	0	A
0	0	1	B
0	1	0	AB
0	1	1	0
1	0	0	$\overline{A}$
1	0	1	$\overline{B}$
1	1	0	$\overline{AB}$
1	1	1	Tercer estado

TABLA 3.25

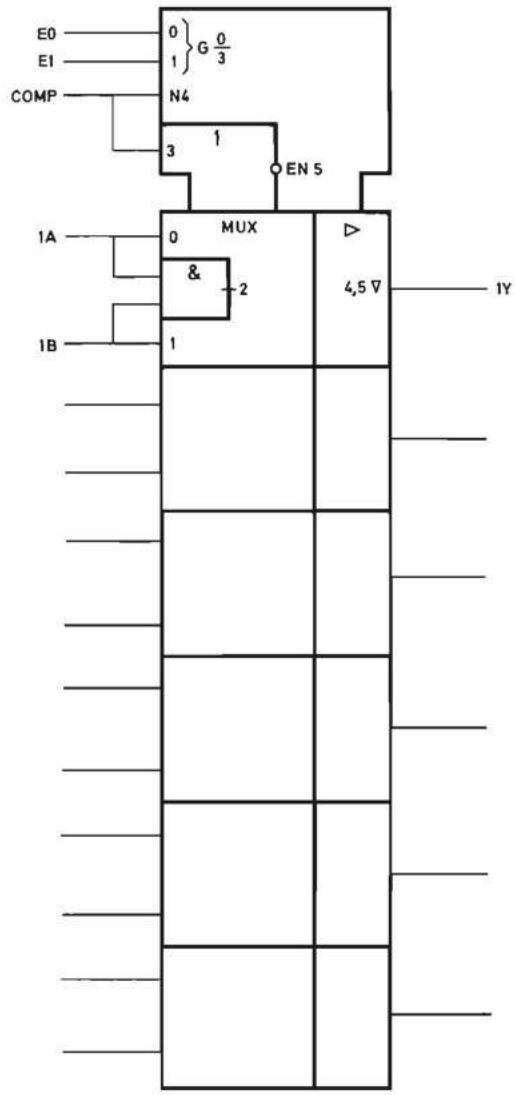


FIGURA 3.83.—**Símbolo de un séxtuple multiplexor de dos canales con función Y incorporada.**

probar que esta tabla se deduce directamente del símbolo lógico normalizado de la figura 3.79*b*.

3.7.3.2 Multiplexores realizados con puertas de tres estados. Las puertas de tres estados son aquellas cuya salida no solamente puede encontrarse en estado cero o uno sino que puede estarlo también en un tercer estado en el que la impe-

$\overline{EN}$	$G\frac{0}{7}$													
I_1	E_2	E_1	E_0	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	Z	$\overline{Z}$
1	X	X	X	X	X	X	X	X	X	X	X	0	1
0	0	0	0	0	X	X	X	X	X	X	X	0	1
0	0	0	0	1	X	X	X	X	X	X	X	1	0
0	0	0	1	X	0	X	X	X	X	X	X	0	1
0	0	0	1	X	1	X	X	X	X	X	X	1	0
0	0	1	0	X	X	0	X	X	X	X	X	0	1
0	0	1	0	X	X	1	X	X	X	X	X	1	0
0	0	1	1	X	X	X	0	X	X	X	X	0	1
0	0	1	1	X	X	X	1	X	X	X	X	1	0
0	1	0	0	X	X	X	X	0	X	X	X	0	1
0	1	0	0	X	X	X	X	1	X	X	X	1	0
0	1	0	1	X	X	X	X	X	0	X	X	0	1
0	1	0	1	X	X	X	X	X	1	X	X	1	0
0	1	1	0	X	X	X	X	X	X	0	X	0	1
0	1	1	0	X	X	X	X	X	X	1	X	1	0
0	1	1	1	X	X	X	X	X	X	X	0	0	1
0	1	1	1	X	X	X	X	X	X	X	1	1	0

X Las entradas correspondientes pueden tomar los valores 0 o 1 lógicos.

TABLA 3.26.—**Tabla de verdad del multiplexor de ocho canales.**

dancia entre ella y los dos terminales de la tensión de alimentación es muy elevada (de varios megohmios).

En el capítulo 5 se estudia la realización de las puertas de tres estados en diversas tecnologías.

Las puertas de tres estados más sencillas son las seguidoras que se representan en la figura 3.84. Estas puertas poseen una entrada de información *D*, una entrada de desinhibición (*EN*) del tercer estado de la salida y una salida. Cuando *EN* se encuentra en nivel uno, la salida sigue a la entrada *D*. Por el contrario cuando *EN* se encuentra en nivel cero, la salida se pone en el tercer estado de alta impedancia.

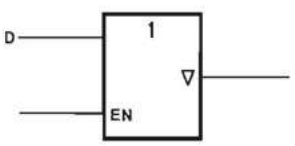


FIGURA 3.84.—**Símbolo lógico de una puerta seguidora de tres estados.**

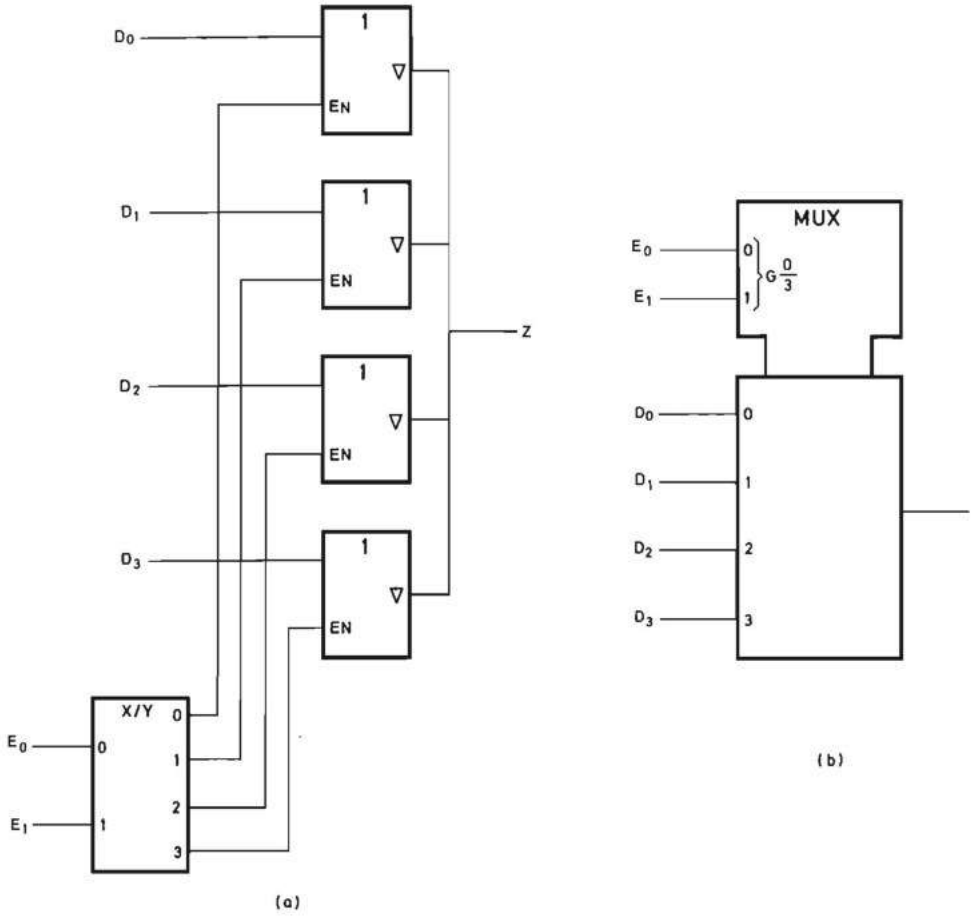


FIGURA 3.85.—**Multiplexor de cuatro canales sin entrada de desinhibición realizado con puertas de tres estados: *a*) Esquema lógico; *b*) Símbolo lógico normalizado.**

En la figura 3.85*a* se representa un multiplexor de cuatro canales realizado con otras tantas puertas seguidoras de tres estados, cuyas salidas se conectan entre sí para obtener la salida del multiplexor. Las entradas de selección E_0 y E_1, se conectan a las entradas de un decodificador de uno entre cuatro, cuyas cuatro salidas se conectan a su vez a las cuatro entradas de desinhibición (*EN*) de las cuatro puertas seguidoras. Se logra de esta forma que en cada instante sólo aparezca en la salida *Z* la información presente en la entrada *D* de la puerta cuya entrada *EN* está en nivel uno. En la figura 3.85*b* se representa el símbolo normalizado.

Dotando al decodificador de una entrada de desinhibición (*EN*) que ponga a cero todas las salidas cuando se encuentra en nivel cero, se obtiene un multiplexor con salida de tres estados. En la figura 3.86*a* y *b* se representan el esquema y el símbolo normalizado de un multiplexor realizado con puertas de tres estados y un decodificador que posee dos entradas de desinhibición I_1 e $\overline{I}_2$ cuyo producto lógi-

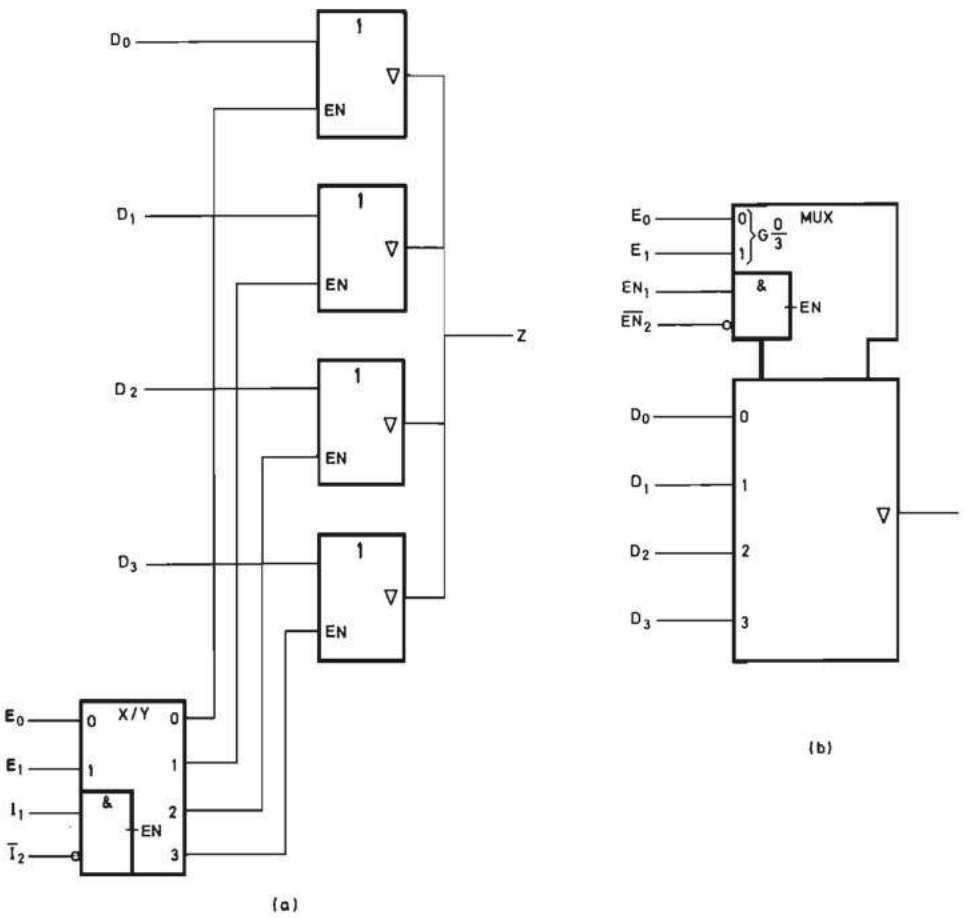


FIGURA 3.86.—**Multiplexor de cuatro canales con entradas de inhibición que controlan el tercer estado de la salida, realizado con puertas de tres estados: *a*) Esquema lógico; *b*) Símbolo lógico normalizado.**

co genera *EN*. Para que el multiplexor esté desinhibido es necesario que simultáneamente I_1 esté en uno e I_2 en cero. Mediante dos multiplexores como el de la figura 3.86 es posible realizar el multiplexor de ocho canales representado en la figura 3.87 que el lector puede comprender fácilmente analizándola con detenimiento.

3.7.3.3 Multiplexores realizados con interruptores. Otra forma de realizar los multiplexores es mediante la conexión de interruptores electrónicos. Un interruptor electrónico es un circuito que posee una entrada, una salida y una variable de control. Cuando la variable de control está en un cierto estado lógico, la resistencia entre el terminal de entrada y el de salida es muy baja (del orden de algunos ohmios) y cuando está en el estado contrario es muy alta (aproximadamente de algunos megohmios).

Existen diversas formas de representar los interruptores electrónicos de las que

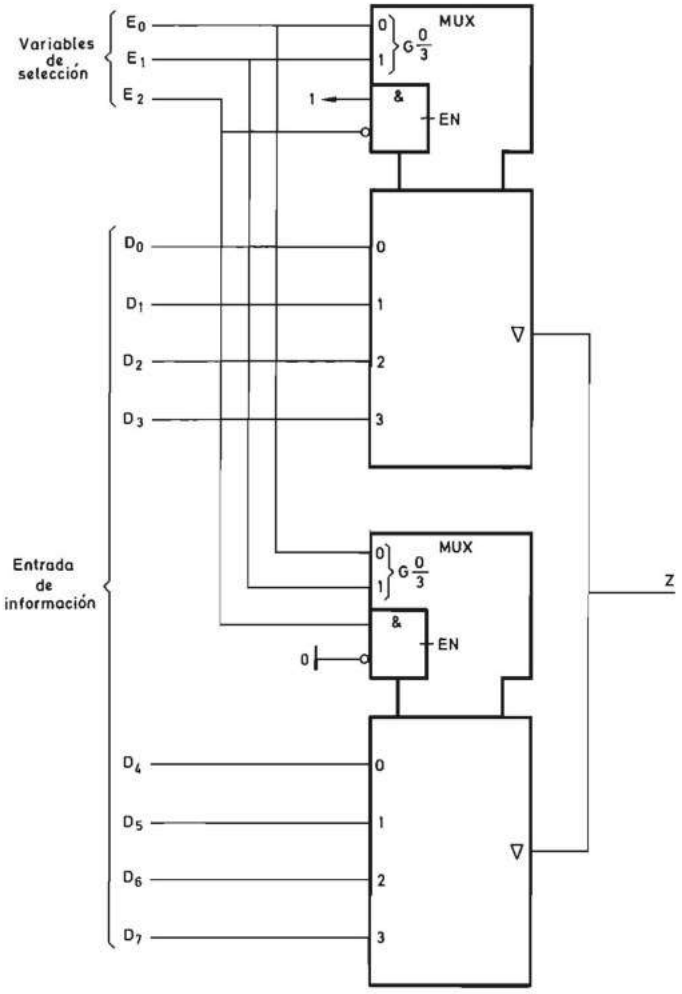


FIGURA 3.87.—**Multiplexor de ocho canales realizado con dos multiplexores como el de la figura 3.86.**

se indican tres en la figura 3.88. En particular es necesario destacar la representada en la figura 3.88*c* que corresponde a la versión normalizada por la Comisión Electrotécnica Internacional. La variable de control recibe la denominación de X que indica precisamente que controla la impedancia entre los otros dos terminales. Para indicar la relación existente se coloca en ambos terminales el dígito 1 y el símbolo X está seguido también de dicho dígito.

En la figura 3.89*a* se representa el esquema de un multiplexor de dos canales de entrada realizado con dos interruptores electrónicos. Los dos terminales de la derecha de cada interruptor se unen entre sí y constituyen la variable de salida del multiplexor. Los dos terminales de la izquierda constituyen las dos variables de entrada del multiplexor. La variable de selección se conecta de forma directa a la en-

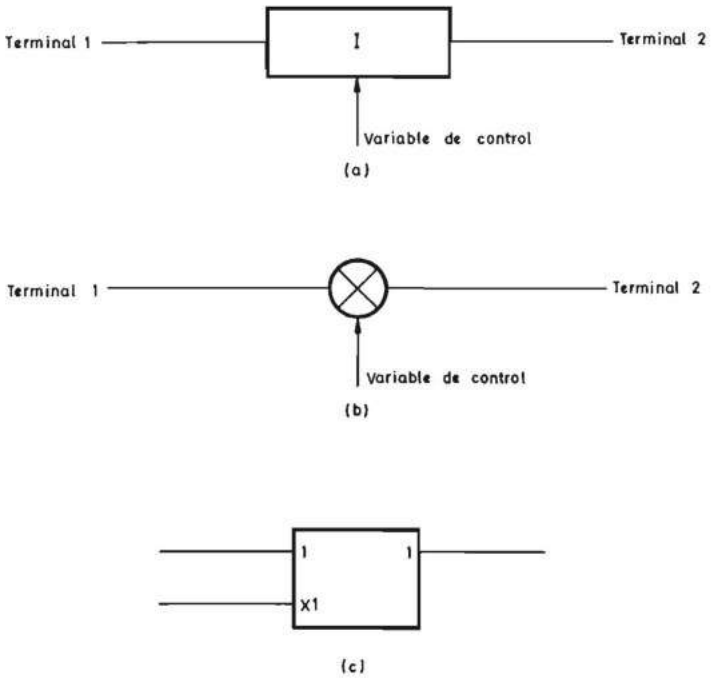


FIGURA 3.88.—**Símbolos lógicos de un interruptor electrónico bidireccional.**

trada $X1$ del interruptor inferior e invertida a la del superior. En la figura 3.89*b* se representa el símbolo normalizado en el que la entrada de selección se denomina $X\ \frac{0}{1}$ y los terminales de entrada 0 y 1. Ello indica que cuando $X\ \frac{0}{1}$ se pone a nivel cero se conecta el terminal de salida al de entrada 0 y por el contrario cuando se pone a uno se conecta al de entrada 1.

La figura 3.90*a* representa un demultiplexor de dos canales, cuyo funcionamiento se recomienda que analice el lector. En la figura 3.90*b* se representa el símbolo normalizado.

Precisamente el hecho de que los dos terminales de cada interruptor sean idénticos hace que los dos esquemas de las figuras 3.89*a* y 3.90*b* lo sean también y que, por lo tanto, constituyan un único circuito que puede ser multiplexor o demultiplexor según por donde se apliquen las señales de entrada y se obtengan las de salida. Se puede, por lo tanto, utilizar un único símbolo lógico para representar a ambos sustituyendo los indicativos MUX y DMUX por el de MDX (figura 3.91).

La principal característica de los circuitos que se acaban de describir es precisamente la de la reversibilidad unida a la de que los interruptores pueden dejar o impedir el paso tanto de señales digitales como analógicas. Por ello, estos circuitos se denominan multiplexores/demultiplexores analógicos.

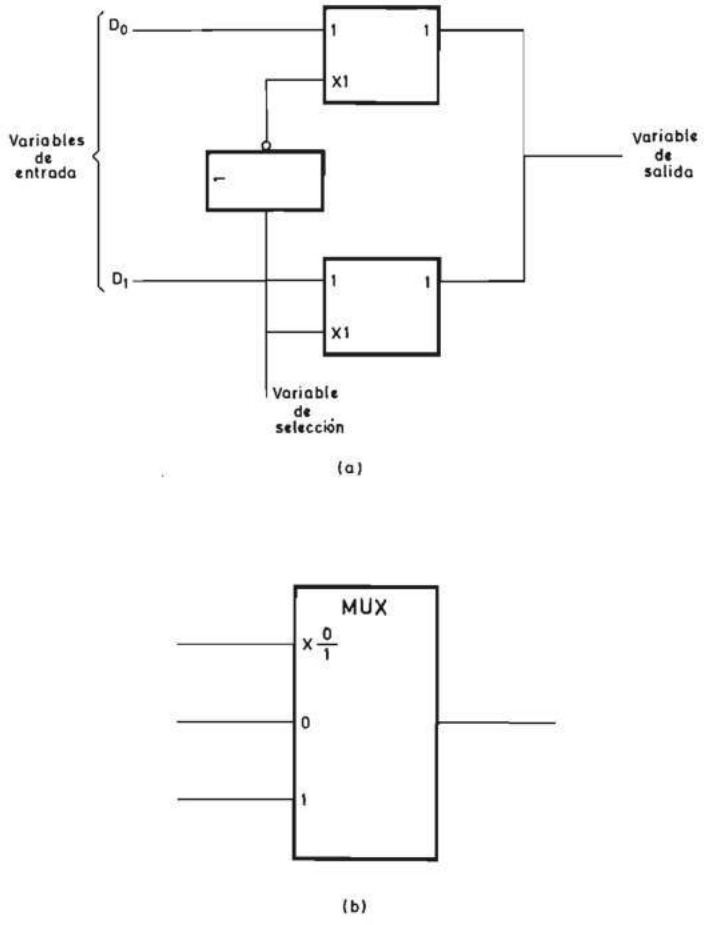


FIGURA 3.89.—**Multiplexor de dos canales realizado con sendos interruptores electrónicos: *a*) esquema lógico; *b*) símbolo normalizado.**

Los multiplexores/demultiplexores analógicos pueden poseer un número cualquiera de canales, en general potencia de dos, y estar dotados de una o más entradas de desinhibición que impidan el cierre del interruptor seleccionado mediante las variables de selección.

En la figura 3.92*a* se representa el esquema de un multiplexor/demultiplexor de cuatro canales. La selección de las variables de control de los interruptores se realiza mediante un decodificador que en este caso es de uno entre cuatro, cuya forma de funcionamiento se analizó en el apartado 3.7.1.

El decodificador de la figura 3.92*a* está dotado de una entrada de desinhibición (*EN*) que, si se encuentra en estado cero, inhibe la activación de la salida correspondiente a la combinación presente en las variables de selección.

En la figura 3.92*b* se representa el símbolo lógico normalizado, en el que se

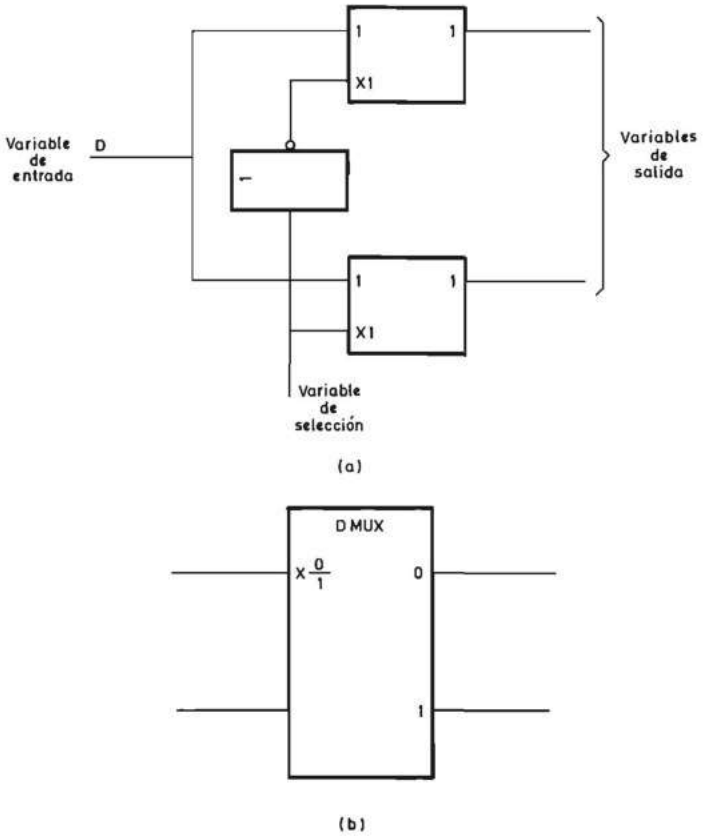


FIGURA 3.90.—**Demultiplexor de dos canales realizado con sendos interruptores electrónicos: *a*) esquema lógico; *b*) símbolo lógico normalizado.**

debe observar que la variable de desinhibición recibe la denominación G4 y las de selección 4 $X\frac{0}{3}$. Los digitos $\frac{0}{3}$ indican que las dos entradas de selección permiten elegir el canal de entrada/salida entre el 0, el 1, el 2 y el 3. El dígito 4 indica

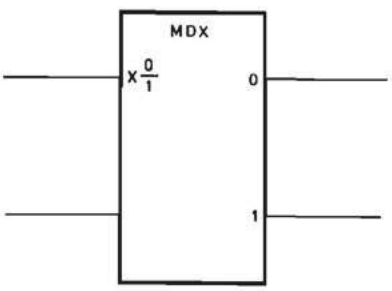


FIGURA 3.91.—**Símbolo lógico normalizado de un multiplexor/demultiplexor de dos canales.**

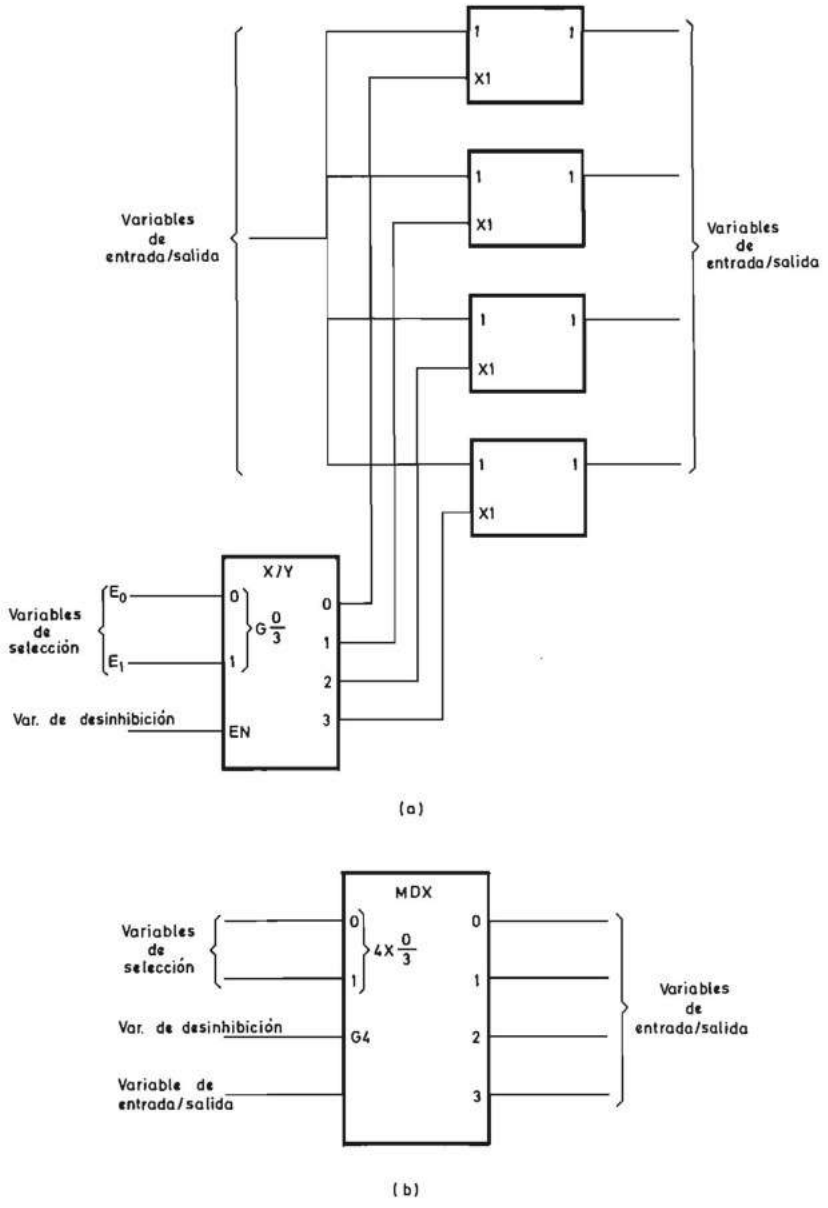


FIGURA 3.92.—**Multiplexor de cuatro canales con entrada de desinhibición: *a*) esquema lógico; *b*) símbolo lógico normalizado.**

que la entrada *G4* inhibe la acción realizada mediante las variables de selección.

En la práctica, la mejor manera de realizar los interruptores electrónicos es mediante transistores MOS y, por ello, los multiplexores/demultiplexores se realizan en tecnología CMOS estudiada en el apartado 5.4.4.4.2.3.

3.7.3.4 Aplicaciones de los multiplexores. Una de las aplicaciones más usuales de los circuitos multiplexores es la de enviar a un solo canal la información procedente de varios canales, seleccionando en cada instante el canal mediante la combinación binaria aplicada a las entradas E.

En la figura 3.93 se representa el esquema de un multiplexor de ocho canales con tres variables de selección E_0, E_1 y E_2. La variable de inhibición I pone a cero la salida cuando se encuentra en nivel uno ($EN = 0$).

La conexión en paralelo de multiplexores con salida de tres estados permite obtener un multiplexor de mayor número de canales. En la figura 3.94 se muestra el circuito correspondiente a un multiplexor de dieciséis canales realizado con dos multiplexores de ocho canales. Para seleccionar los dieciséis canales se necesitan cuatro variables de selección E_0 a E_3. Las tres primeras E_0 a E_2 son comunes a los dos multiplexores (figura 3.94) y la cuarta E_3 se conecta de forma directa a uno de ellos e invertida al otro. Si E_3 se encuentra en nivel cero se pone en tercer estado la salida del multiplexor superior y si se encuentra en nivel uno le sucede lo propio al multiplexor inferior.

Otra aplicación muy interesante de los multiplexores es la generación de funciones lógicas.

Mediante un multiplexor de n variables de selección (2^n canales) se puede generar cualquier función de $n + 1$ variables. Esta afirmación se demuestra con un ejemplo.

En la figura 3.95 se indica la realización de la función de la tabla 3.27 con un multiplexor de dos variables de selección.

Si las variables a y b se conectan a las entradas de selección, cada combinación de ellas hará que a la salida del multiplexor aparezca la información presente en la entrada D correspondiente. Por tanto, ha de seleccionarse la información que debe aparecer en las entradas D del multiplexor, para lo cual se puede hacer uso de la representación tabular de Karnaugh.

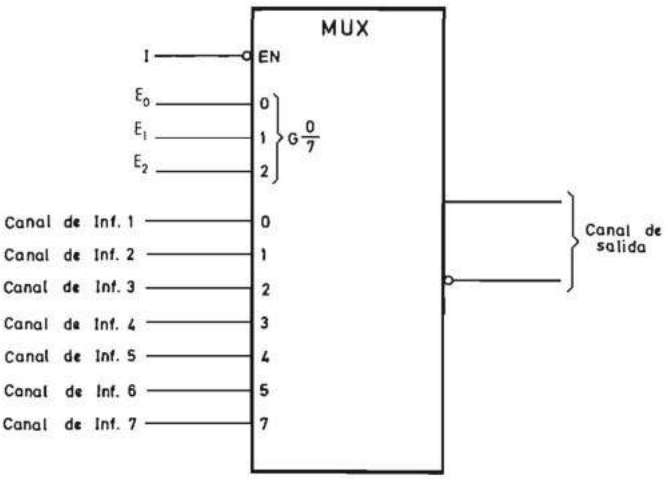


FIGURA 3.93.—**Símbolo lógico de un multiplexor de ocho canales con entrada de desinhibición (*EN*).**

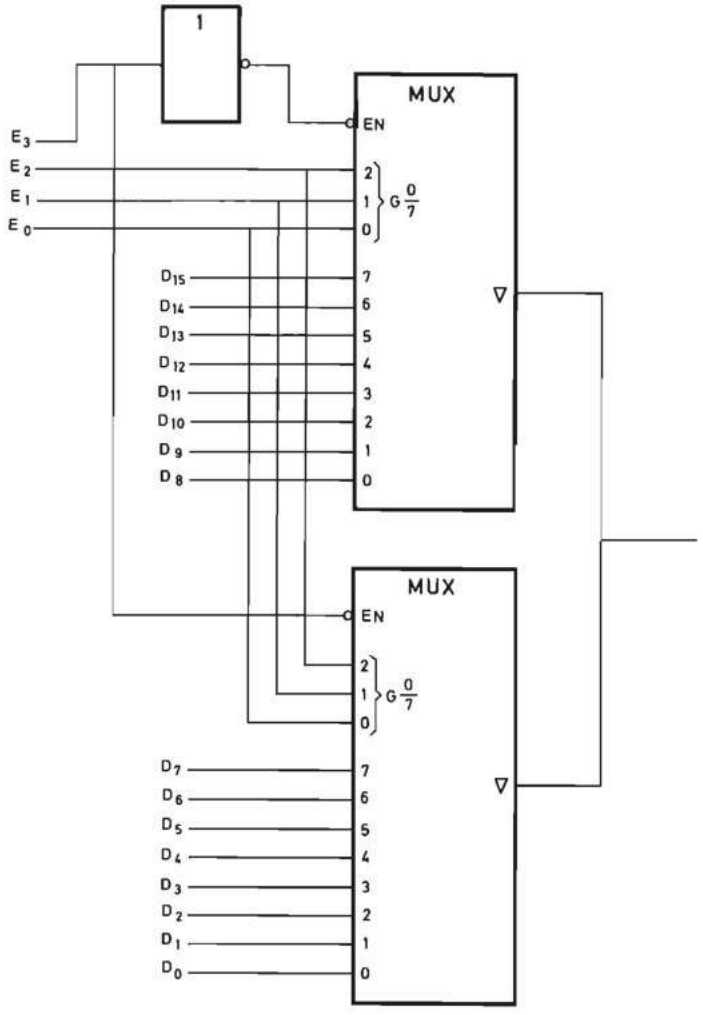
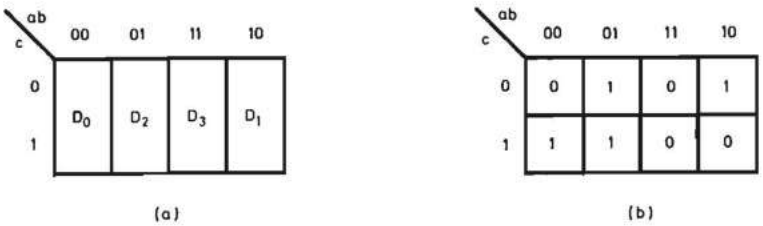


FIGURA 3.94.—**Multiplexor de dieciséis canales realizado con dos multiplexores de ocho canales con salida de tres estados y entrada de desinhibición (*EN*).**

En la figura 3.95*a* se indica para cada combinación de las variables *a* y *b* la entrada *D* cuya información aparece a la salida *Z*. En la figura 3.95*b* se indica el valor que debe tomar *f* para cada combinación de entrada. Si una determinada columna coincide con la variable *c*, ésta ha de ser conectada a la entrada *D* correspondiente. Si, por el contrario, en una columna la función *f* toma los valores inversos de *c*, ha de ser la variable $\bar{c}$ la que se conecte a la entrada correspondiente. Finalmente si una columna es igual a cero o uno, deberá hacerse que la entrada *D* correspondiente sea igual a cero o uno respectivamente. En la figura 3.95*c* se indican los valores que deben tomar las *D* y el esquema lógico de realización de la función *f* con el multiplexor de cuatro canales.



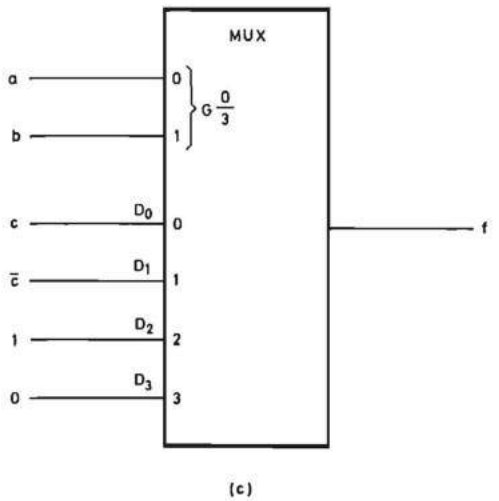


FIGURA 3.95.—**Generación de una función lógica con un multiplexor.**

c	b	a	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

TABLA 3.27

Por ejemplo, cuando a y b toman respectivamente los valores uno y cero lógicos, la función f debe tomar el valor lógico uno si la variable c se encuentra en estado cero y el valor lógico cero en caso contrario. Por tanto, a la entrada D_1 del multiplexor se debe conectar la variable $\bar{c}$.

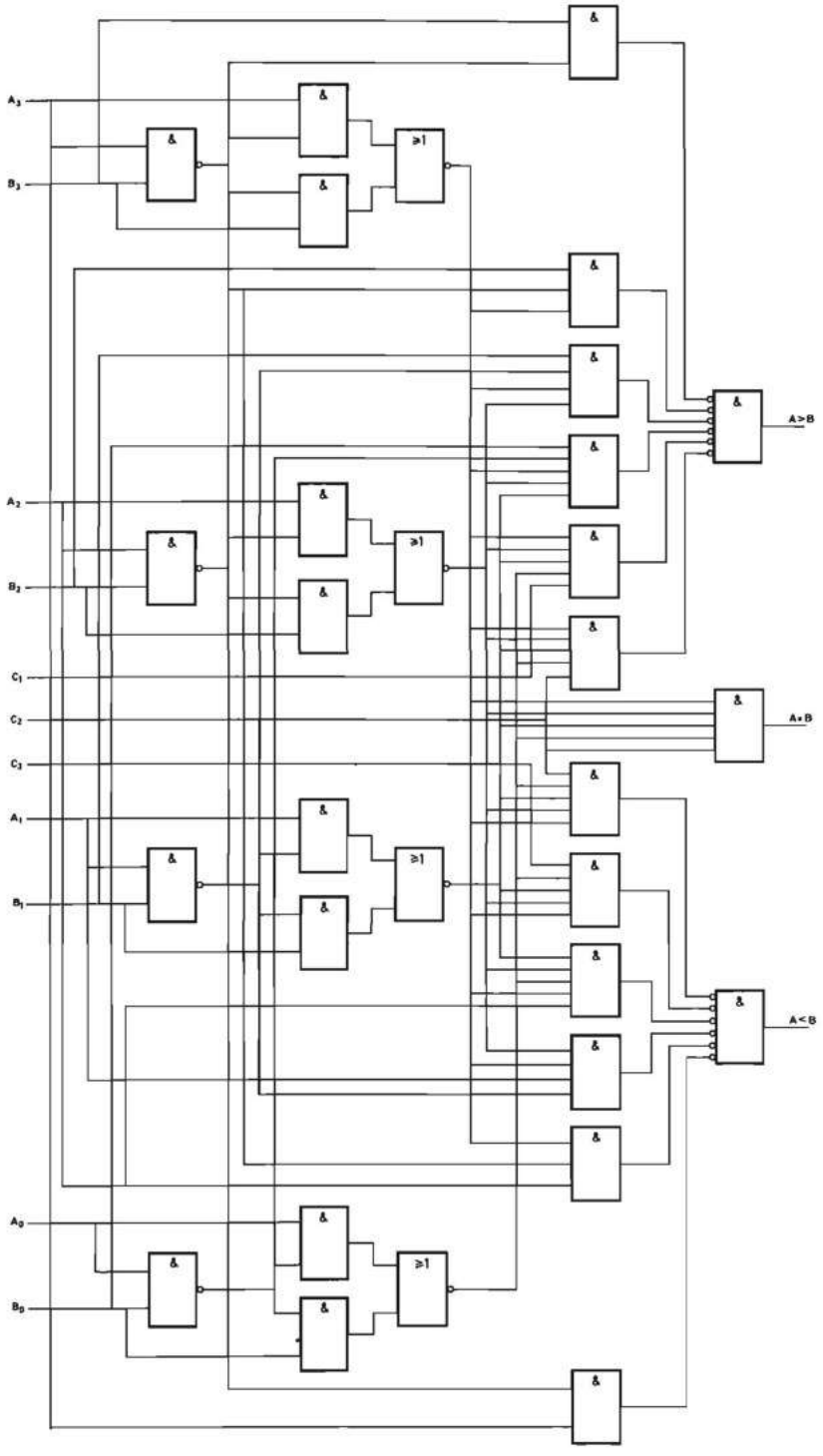


FIGURA 3.96.—**Esquema lógico de un comparador binario de cuatro bits.**

Con un multiplexor de ocho canales se puede generar cualquier función de cuatro variables; con uno de dieciséis cualquier función de cinco variables y mediante combinaciones de todos los tipos de multiplexores descritos se pueden generar todas las funciones lógicas de cualquier número de variables.

3.7.4 Comparadores binarios

Los circuitos comparadores son sistemas combinacionales que detectan si dos combinaciones binarias de n bits en el sistema binario natural son iguales o no y en este último caso cuál de ellas es mayor. Esta función lógica se utiliza con frecuencia en el diseño de sistemas digitales y, por tanto, su disponibilidad como un bloque funcional permite simplificar en gran medida la complejidad de realización de aquéllos.

En escala de integración media (MSI) se han realizado diversos bloques funcionales comparadores.

En la figura 3.96 se representa el esquema de un comparador binario de 4 bits cuya tabla de verdad se indica en la tabla 3.28.

En la figura 3.97 se representan los símbolos no normalizados y normalizados

PQ	$<$	$=$	$>$	$P<Q$	$P=Q$	$P>Q$
$P<Q$	X	X	X	1	0	0
$P>Q$	X	X	X	0	0	0
$P=Q$	1	0	0	1	0	0
$P=Q$	0	1	0	0	1	0
$P=Q$	0	0	1	0	0	1

X Las entradas correspondientes pueden tomar el valor cero o uno lógicos

TABLA 3.28

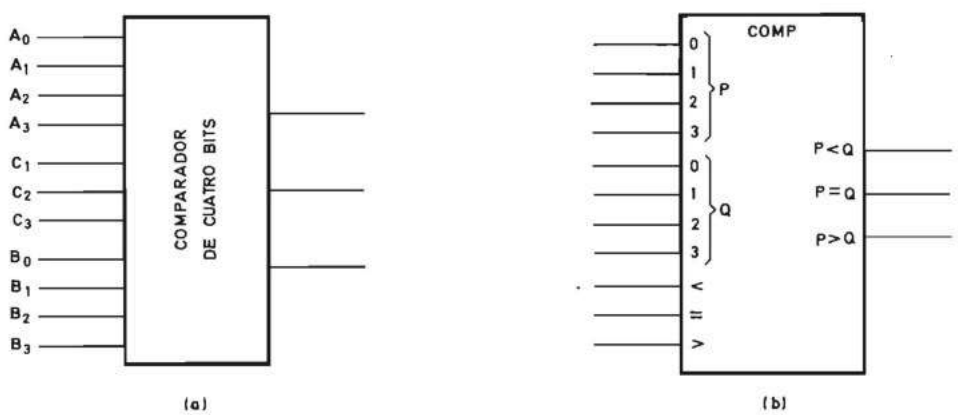


FIGURA 3.97.—**Símbolo lógico del comparador de la figura 3.96: *a*) no normalizado; *b*) normalizado.**

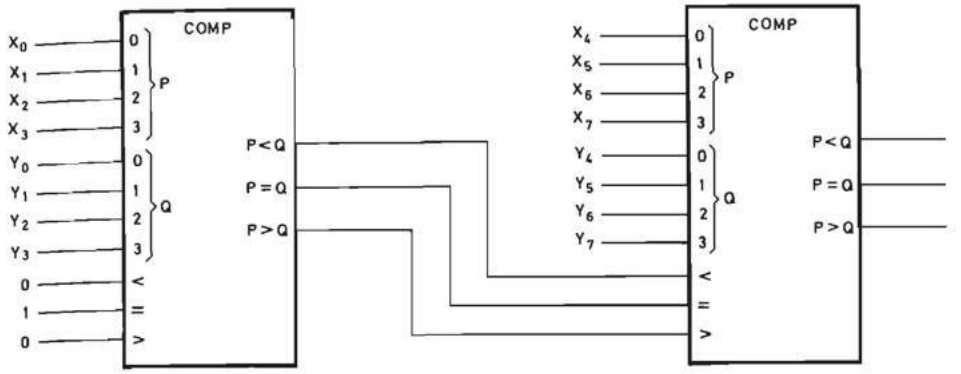


FIGURA 3.98.—**Comparador de ocho bits realizado con el comparador de cuatro cuyo símbolo lógico se representa en la figura 3.97.**

de este comparador que posee tres entradas $<$, $=$ y $>$, que permiten realizar la comparación de combinaciones binarias de un número cualquiera de bits. En la tabla 3.28 se observa que, en el caso de que P y Q sean iguales, el resultado de la comparación depende del valor lógico de $<$, $=$ y $>$ y, por tanto, conectando estas entradas respectivamente a las salidas $P<Q$, $P = Q$ y $P>Q$ de otro comparador y así sucesivamente, se amplía la capacidad de comparación.

En la figura 3.98 se indica el esquema de un comparador de ocho bits realizado con estos circuitos. Primeramente se comparan los cuatro bits menos significativos de cada número y el resultado de esta comparación se conecta a las entradas $<$, $=$ y $>$ de un nuevo comparador a cuyas entradas P_0 a P_3 y Q_0 a Q_3 se conectan los cuatro bits más significativos.

En la figura 3.99 se representa el símbolo lógico normalizado de un comparador de 8 bits que posee una entrada de control $\overline{G}$. La tabla de verdad es la representada en la tabla 3.29. La entrada $\overline{G}$ se comporta como una inhibición/desinhibición que cuando se encuentra en nivel uno fija la salida al nivel cero independientemente de las combinaciones presentes en las entradas P y Q. Por ello en el símbolo normalizado de la figura 3.99 se le asigna el dígito uno tanto a la entrada $\overline{G}$ ($G1$) como a la salida ($1P = Q$). De esta forma se hace superflua la tabla de verdad.

A continuación se analiza un ejemplo de utilización de los circuitos comparadores en combinación con los multiplexores.

Datos P y Q	G	$P = Q$
$P = Q$	0	1
$P > Q$	0	0
$P < Q$	0	0
X	1	0

TABLA 3.29.—**Tabla de verdad del comparador binario de 8 bits de la figura 3.99.**

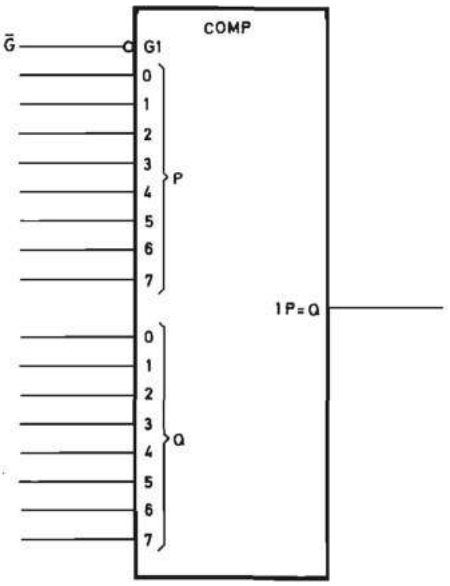


FIGURA 3.99.—**Símbolo lógico normalizado de un comparador de ocho bits con entrada de inhibición /desinhibición (*G*1).**

Ejemplo 3.9: Diseñar un sistema combinacional a cuya entrada se aplican dos combinaciones binarias de cuatro bits y a cuya salida debe aparecer la mayor de ambas y en caso de igualdad, la combinación lógica cero.

En la figura 3.100 se indica la realización práctica de este sistema. Los cuatro bits de ambos números se aplican a las entradas de un comparador de cuatro bits a cuyas salidas se obtiene una indicación de su igualdad o desigualdad y, en este último caso, de cuál es el mayor de ambos. Al mismo tiempo, los bits de igual peso de cada número se llevan a cada uno de los canales de un multiplexor de dos canales. Se necesita, por tanto, un cuádruple multiplexor de dos canales. La decisión del número que aparece a la salida del multiplexor se realiza mediante las entradas de inhibición (*EN*) y de selección (*G*1). La entrada de inhibición *EN* del multiplexor se conecta a la salida $P = Q$ del comparador; cuando ambos números son iguales, esta salida toma el valor uno e inhibe al multiplexor cuyas salidas adoptan todas el estado lógico cero. La salida $P < Q$ del comparador se conecta a la entrada *G*1 de selección del multiplexor. Cuando el número P es menor que el Q, esta salida adopta el valor lógico uno y a las salidas del multiplexor aparece el número Q. Por el contrario, cuando P es mayor que Q esta salida toma el valor lógico cero y, por tanto, a la salida del multiplexor se presenta el número P.

3.7.5 Detectores/generadores de paridad

Este sistema combinacional realiza en esencia la función O-exclusiva de un número de n variables y, en consecuencia, su salida adopta el estado lógico uno si

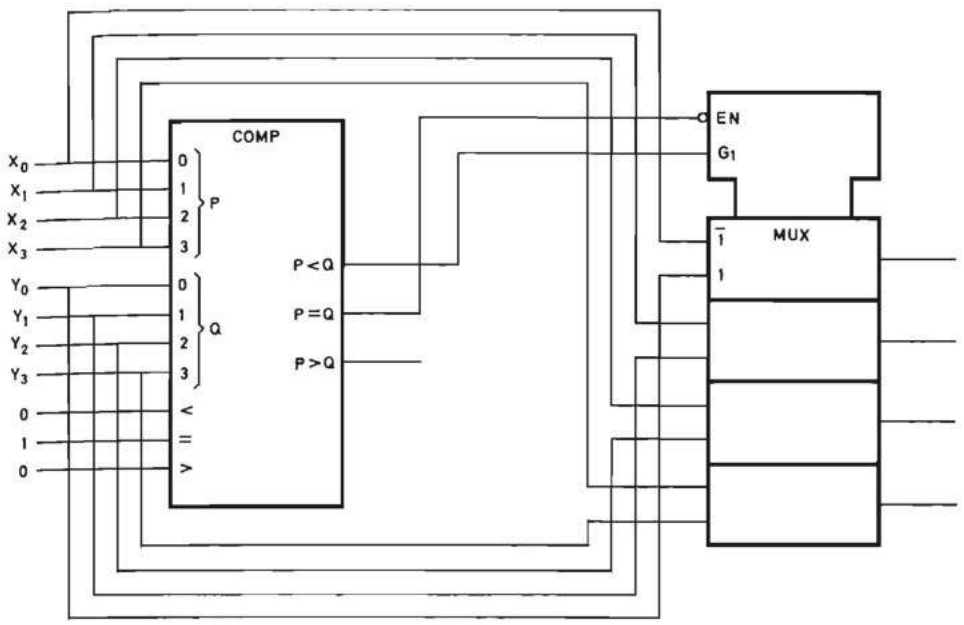


FIGURA 3.100.—**Sistema combinacional que proporciona a su salida el mayor de dos números de cuatro bits, y un cero si son iguales.**

el número de unos aplicados a sus entradas es impar y un cero lógico si dicho número es par. Por tanto, este circuito permite detectar la paridad de una combinación binaria y, si se añade el bit obtenido a la combinación de n bits, se logra una combinación de $n + 1$ bits cuya paridad es constante.

La aplicación más importante de este circuito es la detección de errores en códigos detectores y correctores de error, así como la generación de estos códigos.

En la figura 3.101 se representa el esquema de un detector/generador de paridad de 9 bits realizado en escala de integración media cuya tabla de verdad se indica en la tabla 3.30.

El lector puede comprobar que la ecuación lógica de las salidas *PI* (paridad impar) y *PP* (paridad par) es:

$$PI = A \oplus B \oplus C \oplus D \oplus E \oplus F \oplus G \oplus H \oplus I$$

$$PP = \overline{PI}$$

PI adopta el estado lógico uno si se encuentra en dicho estado un número impar de las variables *A* a *I*.

En la figura 3.102 se representa el símbolo lógico normalizado de este circuito.

Mediante el acoplamiento en serie de dos circuitos de este tipo se consigue un detector/generador de paridad de 17 bits cuyo esquema se representa en la figura 3.103.

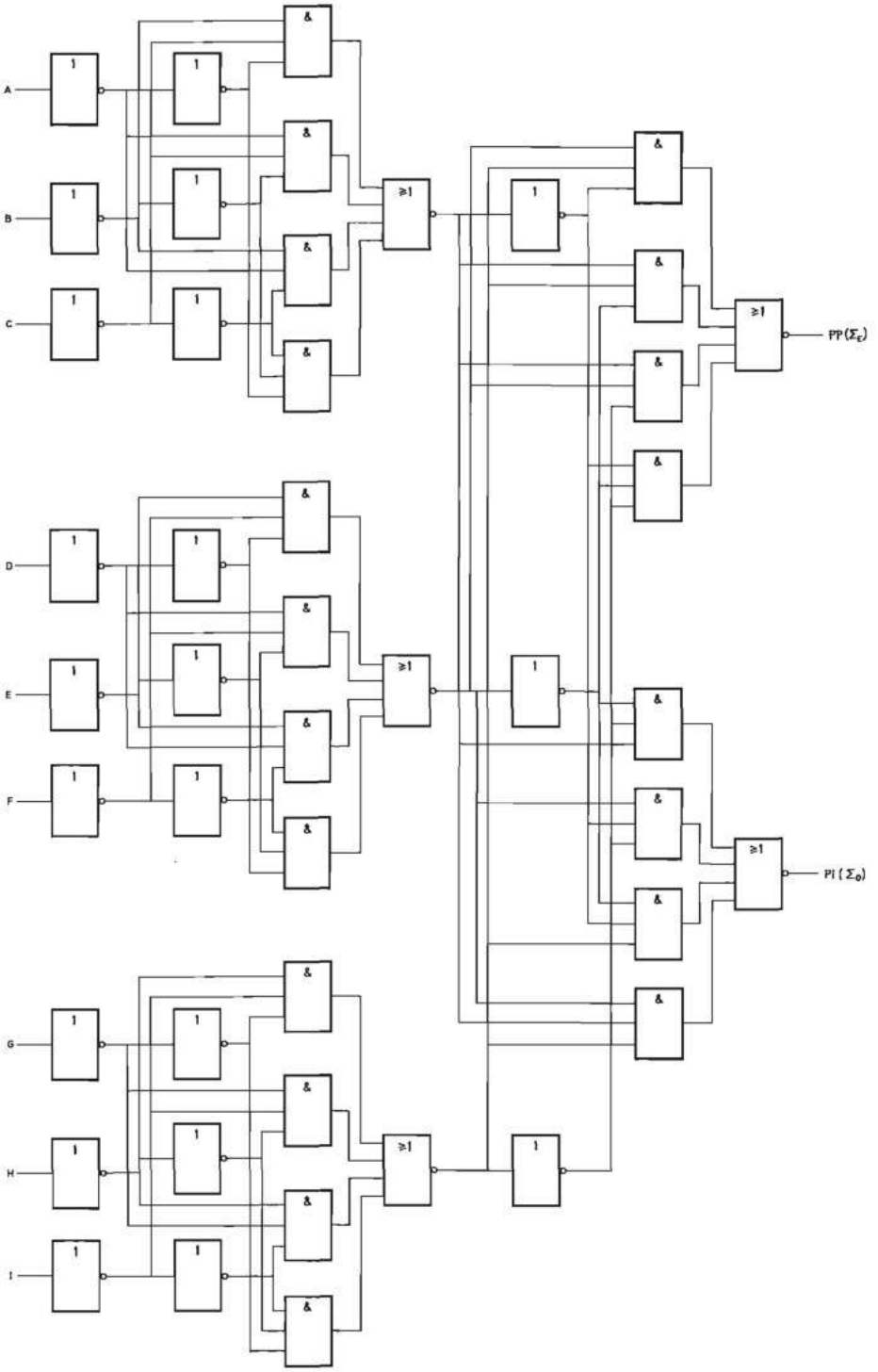


FIGURA 3.101.—**Esquema lógico de un detector/generador de paridad de nueve bits.**

I H G F E D C B A	*PI*	*PP*
Todas las entradas en estado cero	0	1
Una entrada cualquiera en estado uno	1	0
Dos entradas cualesquiera en estado uno	0	1
Tres entradas cualesquiera en estado uno	1	0
Cuatro entradas cualesquiera en estado uno	0	1
Cinco entradas cualesquiera en estado uno	1	0
Seis entradas cualesquiera en estado uno	0	1
Siete entradas cualesquiera en estado uno	1	0
Ocho entradas cualesquiera en estado uno	0	1
Nueve entradas cualesquiera en estado uno	1	0

TABLA 3.30

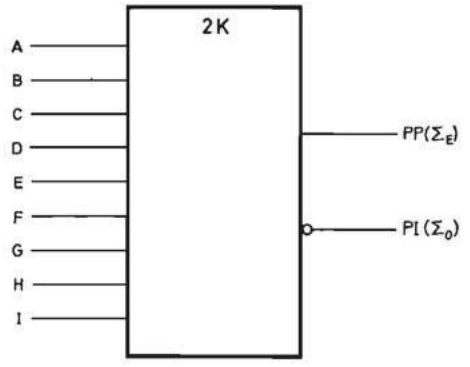


FIGURA 3.102.—**Símbolo lógico del detector/generador de paridad de la figura 3.101.**

3.8 SISTEMAS COMBINACIONALES PROGRAMABLES

Se pueden definir los sistemas combinacionales programables como aquellos cuya tabla de verdad puede ser cambiada sin necesidad de modificar el cableado entre los elementos que los constituyen.

El progreso de las técnicas de integración ha permitido la realización física de sistemas combinacionales programables de elevado número de variables de entrada y salida en escala de integración media (MSI) y gran escala de integración (LSI).

Los sistemas combinacionales programables se pueden clasificar en dos grandes tipos según se indica en la tabla 3.31:

a) Sistemas combinacionales programables no universales que realizan funciones específicas de aplicación general.

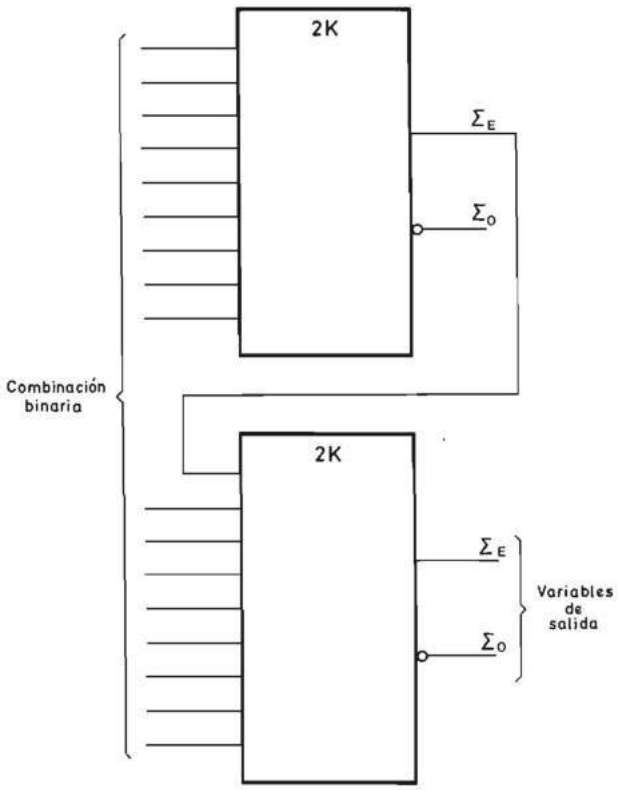


FIGURA 3.103.—**Detector/generador de paridad de 17 bits realizado con el detector/generador de paridad cuyo símbolo lógico se representa en la figura 3.102.**

b) Sistemas combinacionales programables universales con los que se puede realizar cualquier función lógica.

La programación consiste en la supresión de determinadas conexiones realizadas mediante un diodo o un transistor. Inicialmente se utilizaron diodos y la programación consistía en hacer pasar a través de ellos una corriente superior a la de funcionamiento normal que los fundía. El desarrollo de los transistores MOS ha hecho que se realicen sistemas combinacionales programables cuyas conexiones están constituidas por transistores MOS de puerta flotante que se hacen conducir o no mediante la aplicación de una tensión superior a la de funcionamiento normal que inyecta portadores de carga a la puerta durante el proceso de programación. En el apartado 7.3.2.2 se describen las memorias pasivas que utilizan estas técnicas. Al lector que desee estudiarlas en mayor profundidad se le remite a la bibliografía [INTE 89] [PHIL 89] [CYPR 89].

En sucesivos apartados se estudian los diferentes tipos de sistemas combinacionales programables.

TABLA 3.31

3.8.1 Sistemas combinacionales programables no universales

Existen algunas funciones lógicas que, aunque no son universales, se utilizan en numerosas aplicaciones y presentan variantes cuya programación resulta interesante desde un punto de vista práctico.

A continuación se estudian dos sistemas combinacionales programables que realizan funciones de ese tipo.

3.8.1.1 Matrices programables de puertas Y o decodificadores programables [Programmable gate arrays (PGA)]. Su esquema se representa en la figura 3.104*a* en la que se observa que están formadas por un cierto número n' de puertas Y conectadas a un número n de variables de entrada y sus inversas. El número n' de puertas es menor que 2^n y mediante la supresión de las conexiones adecuadas se logra que la salida de cada una de ellas constituya un producto canónico entre los 2^n posibles. De lo expuesto se deduce que este circuito constituye. un decodificador programable. El esquema de la figura 3.104*a* se puede representar de forma simplificada tal como se indica en la figura 3.104*b* en la que todas las conexiones de cada puerta Y se indican en una sola línea. En la figura 3.105 se representa este circuito mediante símbolos normalizados.

3.8.1.2 Detectores de identidad (Identity comparators). Su esquema básico se representa en la figura 3.106*a* y están constituidos por un comparador, una de cuyas combinaciones de entrada es programable, es decir, sus bits se pueden colocar en cero o en uno mediante la supresión de la conexión adecuada. En efecto, cada uno de los bits de la combinación Q de la figura 3.106*a* está a cero si el

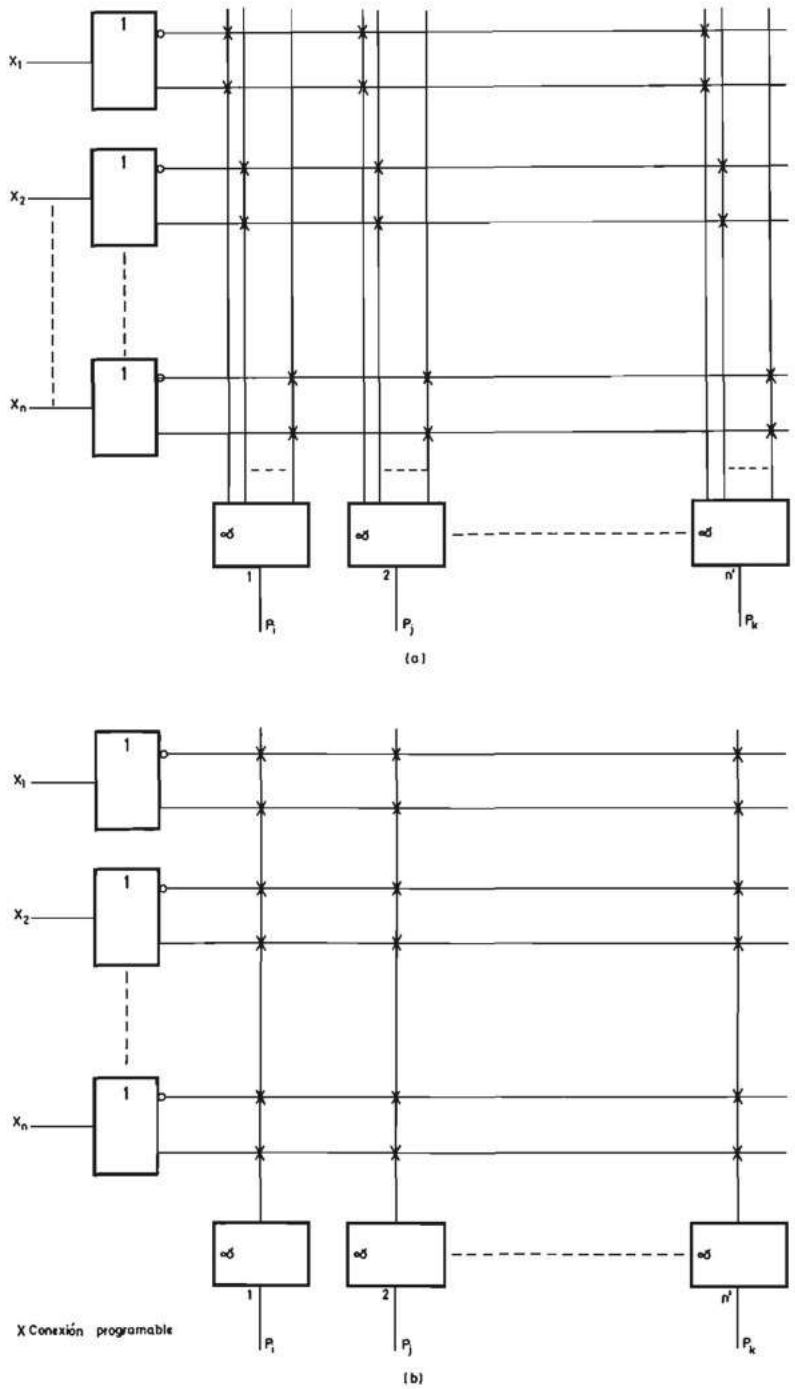


FIGURA 3.104.—**Esquema lógico de una matriz programable de puertas Y [Programmable gate array (PGA)]: *a*) no simplificado; *b*) simplificado.**

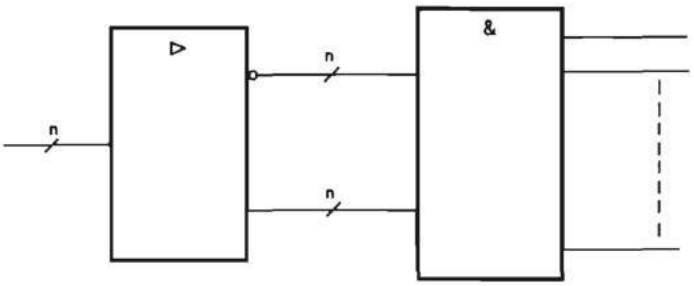
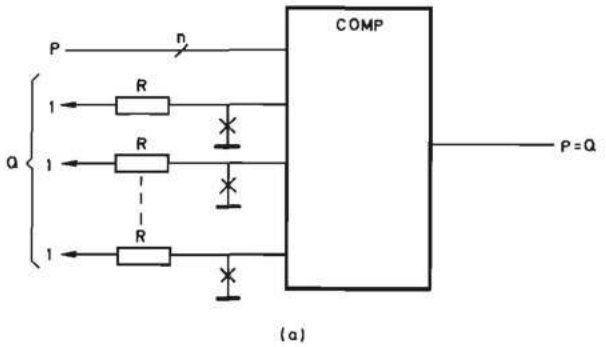
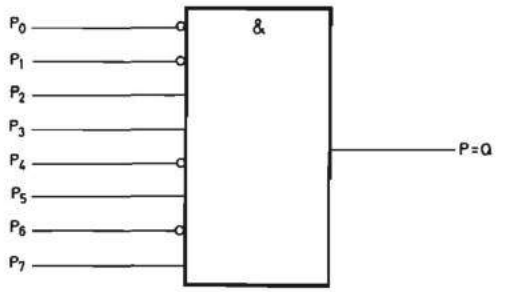


FIGURA 3.105.—**Esquema de una matriz programable de puertas Y realizado con símbolos lógicos normalizados.**

elemento (diodo, transistor, etc.) representado mediante el símbolo X conduce y está a uno si por el contrario está cortado. La salida $P = Q$ se pone a nivel uno cuando la combinación presente en la entrada P coincide con la programada en la entrada Q. Este circuito, por lo tanto, identifica la combinación presente en Q y tiene como principal aplicación la selección de unidades de entrada o salida modulares de un procesador digital [MAND 92].



(a)



(b)

FIGURA 3.106.—**Detector de identidad (Identity comparator): *a*) esquema básico; *b*) símbolo lógico.**

En la figura 3.106*b* se representa el símbolo lógico de este bloque funcional una vez programado. La entrada *Pi* cuya homóloga *Qi* ha sido programada en uno no lleva el símbolo de inversión de entrada y, por el contrario, sí lo lleva aquella que ha sido programada en cero.

El circuito de la figura 3.106*b* detecta, por lo tanto, la combinación 10101100.

La forma de programación depende de la tecnología y la realización física utilizada por el fabricante de circuitos integrados.

El lector debe por ello estudiar los manuales de datos [TEXA 85] para diseñar sistemas con estos componentes.

3.8.2 Sistemas combinacionales universales programables

Estos sistemas se pueden a su vez clasificar en completos o incompletos, según sea o no posible programar el valor de las variables de salida para cada combinación de las variables de entrada de forma independiente. A continuación se estudia cada uno de estos tipos.

3.8.2.1 Sistemas combinacionales universales programables completos. Memorias de acceso aleatorio. Se definen como sistemas combinacionales completos aquellos en los que es posible programar de forma independiente el valor de las variables de salida correspondiente a cada una de las combinaciones de las variables de entrada. Las memorias de acceso aleatorio [en inglés «Random Access Memories» (RAM)] en sus diferentes versiones constituyen dispositivos lógicos programables combinacionales universales completos. Debido a que las memorias de acceso aleatorio forman parte de la práctica totalidad de los procesadores digitales secuenciales, a su estudio se dedica el capítulo 7 de este libro. No obstante, a continuación se realiza una breve introducción a este tipo memorias y para un estudio más profundo se remite al lector al citado capítulo.

Una memoria de acceso aleatorio (RAM) está constituida por un cierto número N de células capaces de almacenar una información binaria (0 o 1) agrupadas en posiciones de m células de manera que el número total N' de posiciones cumple la ecuación $N = N' \cdot m$. La memoria posee en el caso más general m terminales de entrada cuya información puede ser introducida en las m células de cualquier posición en una operación de escritura y m terminales de salida en los que puede aparecer la información de las m células de cualquier posición en una operación de lectura. Ambos grupos de terminales se pueden confundir en uno solo, utilizado indistintamente para introducir información en la memoria o leer la que contiene ésta. En la figura 3.107 se representan los símbolos lógicos no normalizado (*a*) y normalizado (*b*) de una memoria de acceso aleatorio (RAM).

Para poder seleccionar cuál de las N' posiciones se lee o escribe, la memoria posee n terminales de dirección tales que $2^n = N'$. Cada una de las 2^n combinaciones posibles de las n variables de dirección selecciona una de las N' posiciones de la memoria.

El lector puede comprender fácilmente que una memoria de acceso aleatorio constituye un sistema combinacional programable. En efecto, una memoria de ac-

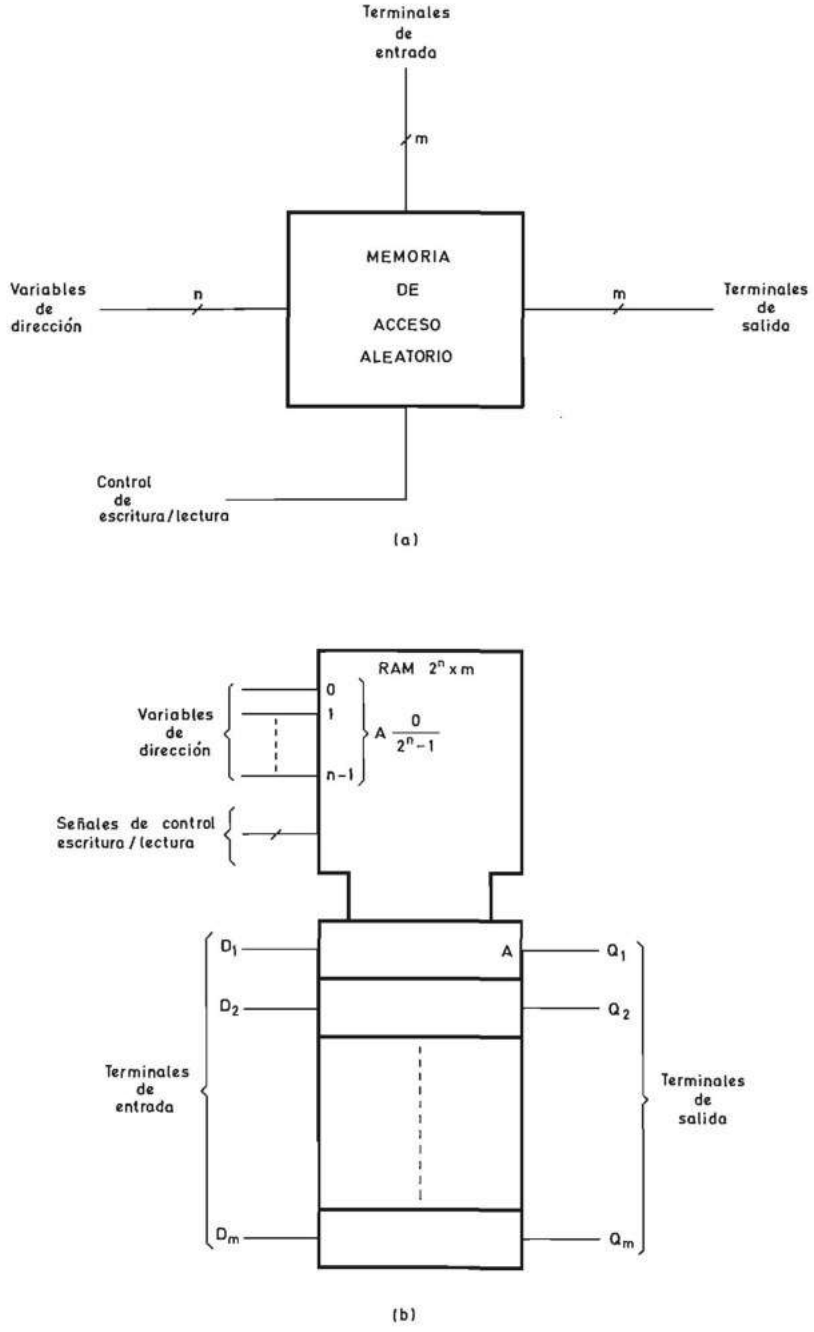
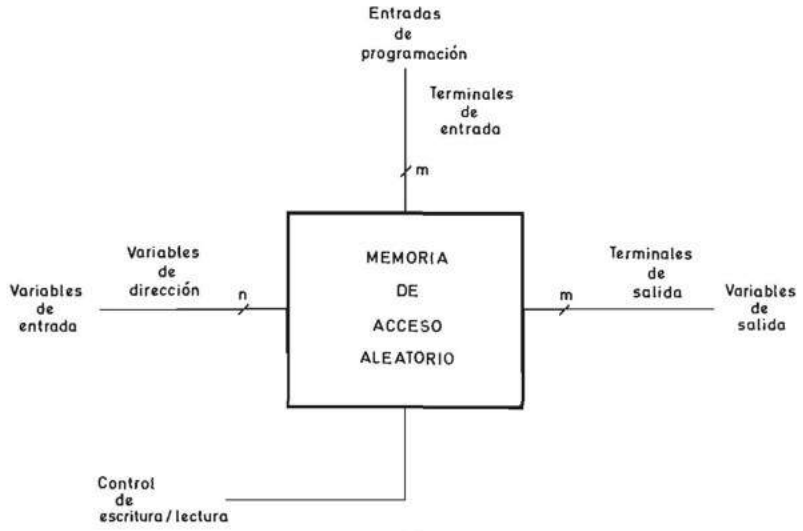
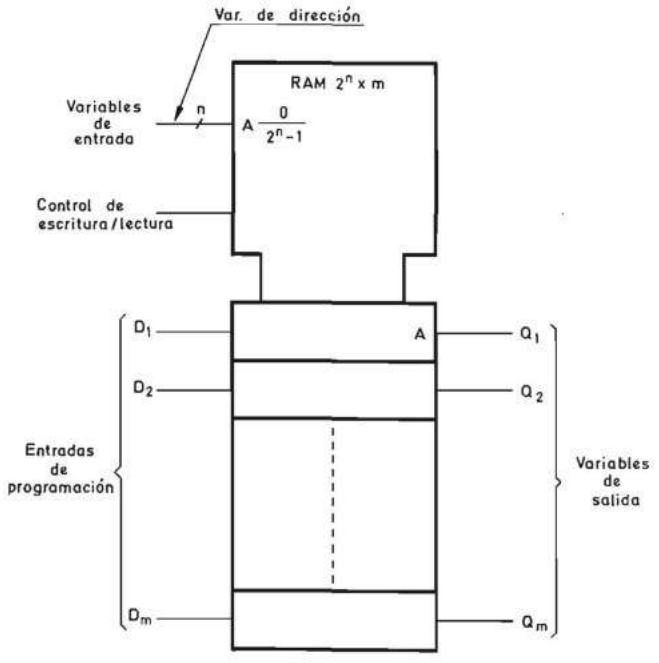


FIGURA 3.107.—**Memoria de acceso aleatorio: *a*) símbolo lógico no normalizado; *b*) símbolo lógico normalizado.**



(a)



(b)

FIGURA 3.108.—**Memoria de acceso aleatorio utilizada como circuito combinacional programable completo.**

ceso aleatorio se comporta como un sistema combinacional durante la operación de lectura, porque para cada combinación binaria aplicada a sus *n* entradas de dirección aparece una información de salida igual a la contenida en la posición seleccionada, que es independiente de la secuencia de combinaciones de las variables de entrada. Las entradas de dirección de la memoria constituyen las variables de entrada del sistema combinacional y las salidas de información constituyen las variables de salida, tal como se indica en la figura 3.108. Pero además este sistema combinacional es programable, porque mediante operaciones de escritura se puede modificar la información contenida en cada posición introduciendo en ella la información colocada en los terminales de entrada, que constituyen las entradas de programación del circuito (fig. 3.108). La señal de «control de la escritura/lectura» permite seleccionar mediante su nivel lógico la operación de escritura o lectura.

Por lo tanto, una memoria de acceso aleatorio se comporta durante la operación de lectura como un sistema combinacional, tal como se indica de forma gráfica en la figura 3.109 en la cual se prescinde de la señal de «control de escritura/lectura» que se supone conectada rígidamente al nivel lógico correspondiente a la operación de lectura. Los terminales de entrada se utilizan exclusivamente para realizar la programación mediante operaciones de escritura en las diferentes posiciones de la memoria.

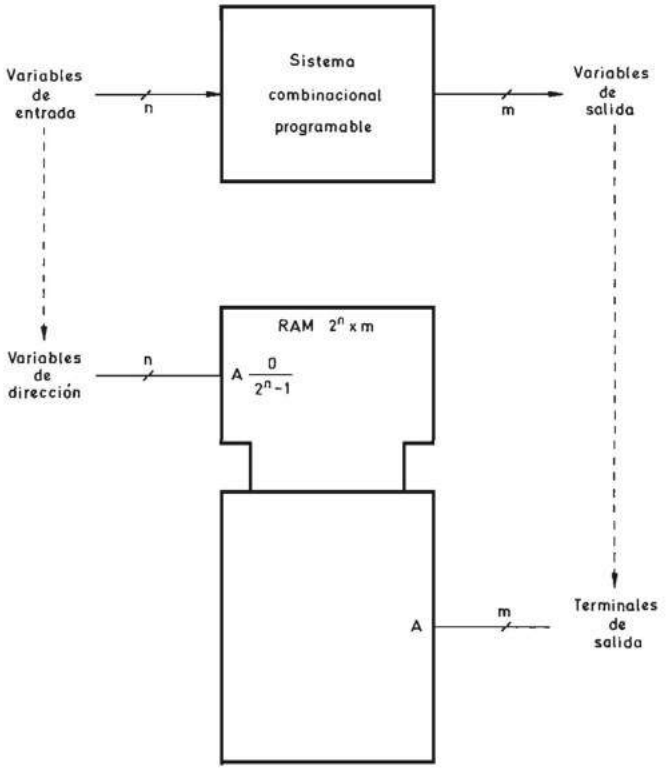


FIGURA 3.109.—**Memoria de acceso aleatorio utilizada como circuito combinacional.**

Mediante un ejemplo se aclara todo lo que se acaba de exponer.

Ejemplo 3.10: Realizar mediante una memoria de acceso aleatorio el sistema combinacional cuya tabla se representa en la tabla 3.32.

Es necesario deducir la organización de la memoria, es decir el número de posiciones y el número de bits de cada posición y el tipo de memoria de acceso aleatorio.

El número de posiciones ha de ser igual al de combinaciones de las variables de entrada, es decir, $2^4 = 16$. Cada posición ha de tener dos bits correspondientes a las dos variables de salida f_1 y f_2. Por ello, la memoria ha de poseer cuatro entradas de dirección y dos variables de salida y su símbolo lógico se representa en la figura 3.110. Las variables de entrada a, b, c y d se conectan respectivamente a las entradas de dirección A_0, A_1, A_2 y A_3 y los terminales de salida S_1 y S_2 coinciden con las variables de salida f_1 y f_2 respectivamente.

En cada una de las posiciones de la memoria se ha de colocar la información de f_1 y f_2 correspondiente a cada combinación de las variables de entrada indicada en la tabla 3.32.

En el capítulo 7 se estudian los diferentes tipos de memorias de acceso aleatorio (RAM) y se demuestra que las más adecuadas para la realización de sistemas combinacionales son las pasivas en sus diferentes versiones debido a su característica de no perder la información al suprimir la tensión de alimentación.

En este ejemplo se ha elegido una memoria pasiva programable (PROM) (figura 3.110).

d	c	b	a	f_1	f_2
0	0	0	0	1	1
0	0	0	1	0	0
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	1
1	1	1	1	0	0

TABLA 3.32

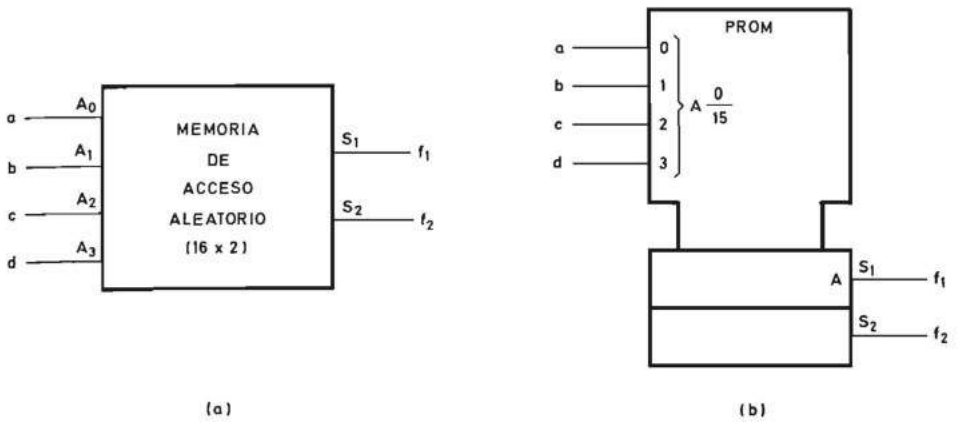


FIGURA 3.110.—**Realización con una memoria de acceso aleatorio del circuito combinacional cuya tabla de verdad se representa en la tabla 3.32: *a*) símbolo no normalizado; *b*) símbolo normalizado.**

El símbolo lógico normalizado asignado a las memorias pasivas es idéntico al de las memorias de acceso aleatorio representado en la figura 3.107. En la figura 3.111*a* se representa el símbolo correspondiente a una memoria totalmente pasiva (ROM) de 2^n posiciones de m bits con salida de tres estados controlada mediante

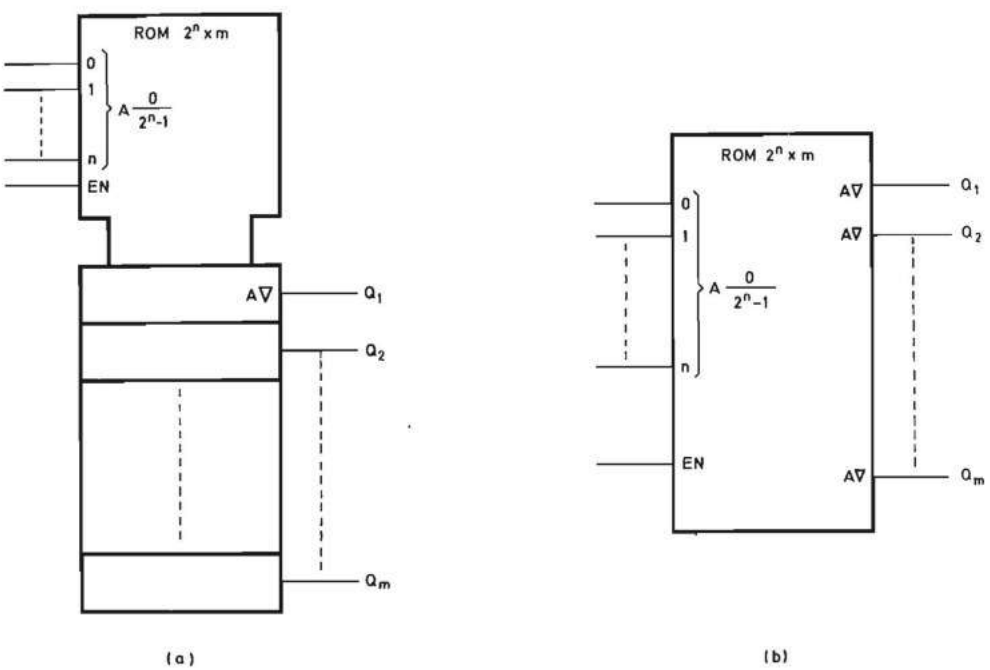


FIGURA 3.111.—**Símbolos lógicos normalizados de una memoria de acceso aleatorio totalmente pasiva.**

la variable de entrada *EN*. En la figura 3.111*b* se representa un símbolo también normalizado alternativo.

Las memorias de acceso aleatorio pasivas pueden tener una organización 2D, 3D o 2½ D (ver apartado 7.2.3.1.2 del capítulo 7). La organización 2D se representa en la figura 3.112. El decodificador selecciona mediante cada una de sus variables de salida las distintas posiciones. El decodificador se puede realizar mediante una matriz de puertas Y adecuadamente conectadas a las variables de entrada de tal manera que cada puerta Y decodifica un producto canónico. Se tiene así una matriz de $n\,2^n$ conexiones fijas (figura 3.113).

Las salidas de las puertas Y constituyen las columnas de otra matriz cuyas filas se conectan a un número m de puertas O. Las puertas O se conectan a las salidas de todas las puertas Y (figura 3.113). La complejidad de la figura 3.113 lleva a la representación simplificada de la figura 3.114 de la que se deduce que una memoria pasiva (ROM, PROM o RPROM) está constituida por una matriz fija de puertas Y y una matriz programable de puertas O.

En la figura 3.115 se representa la realización de las funciones f_1 y f_2 de la tabla 3.32 mediante una memoria pasiva. En la matriz de puertas O se eliminan las conexiones correspondientes a los productos canónicos que no forman parte de cada una de las funciones.

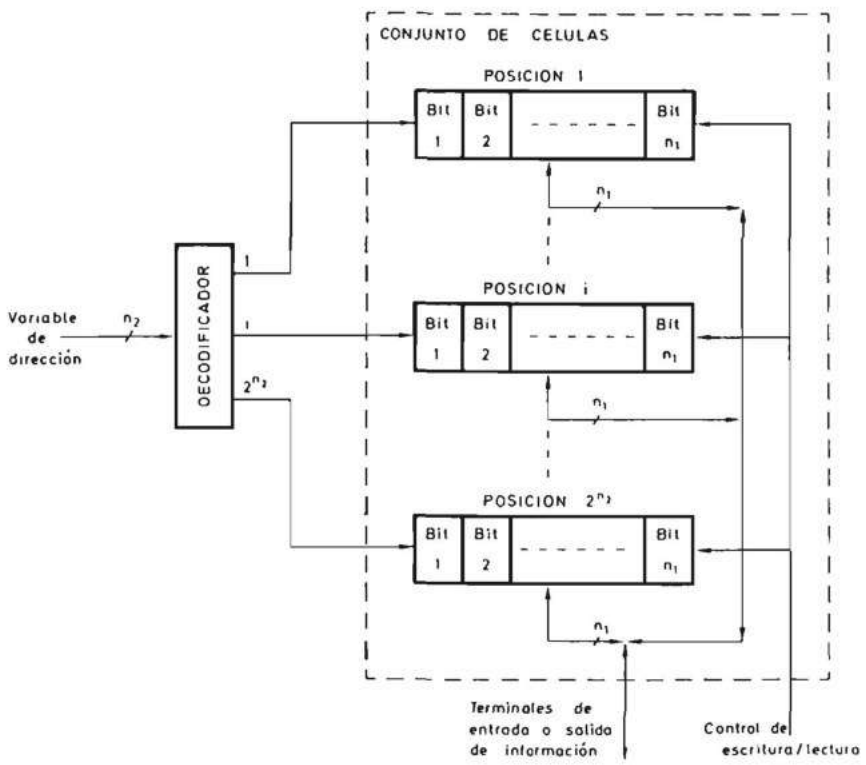


FIGURA 3.112.—**Organización 2D de una memoria de acceso aleatorio.**

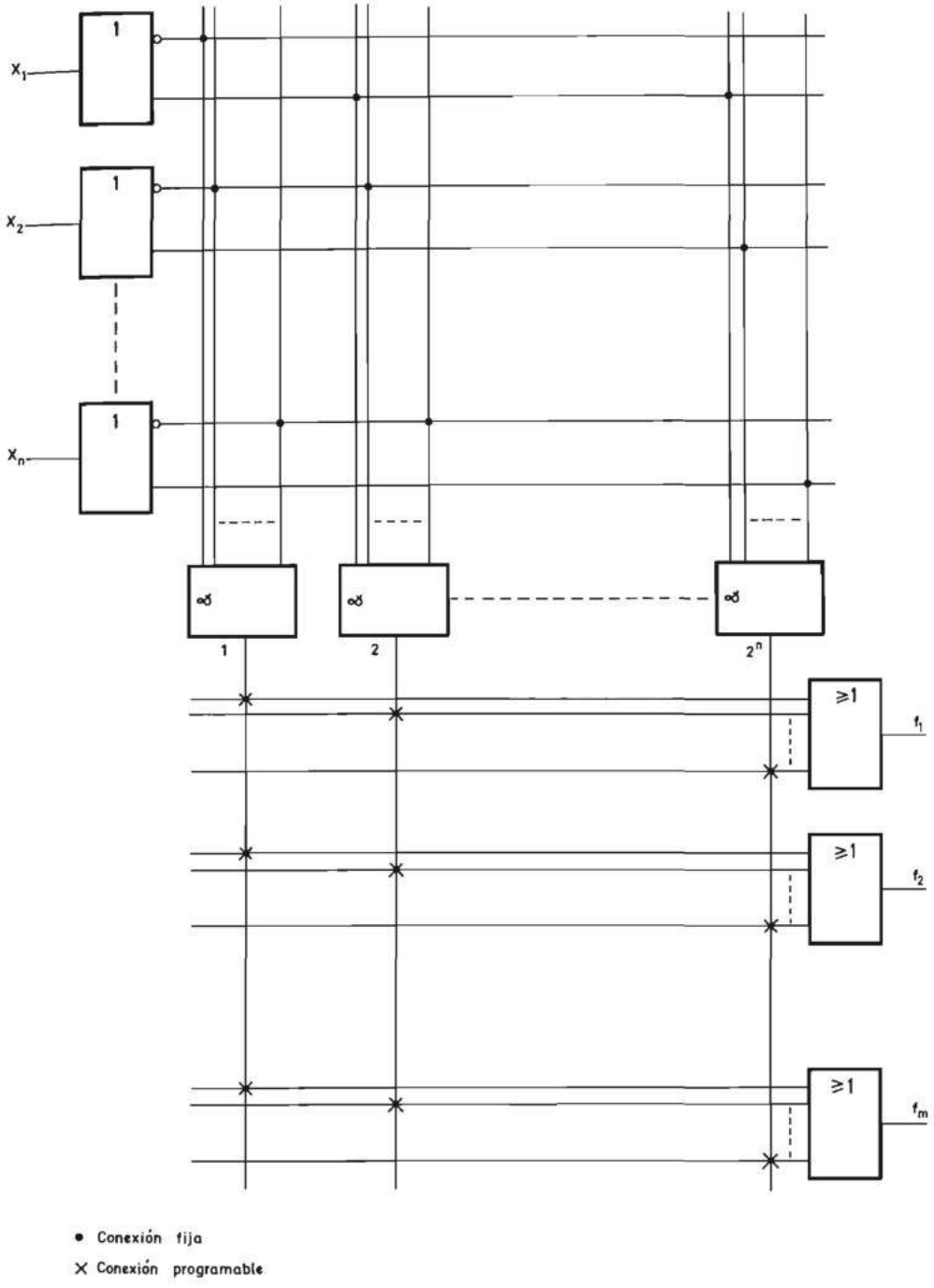


FIGURA 3.113.—**Esquema de una memoria pasiva (ROM, PROM o RPROM) realizada mediante una matriz de puertas Y y otra de puertas O.**

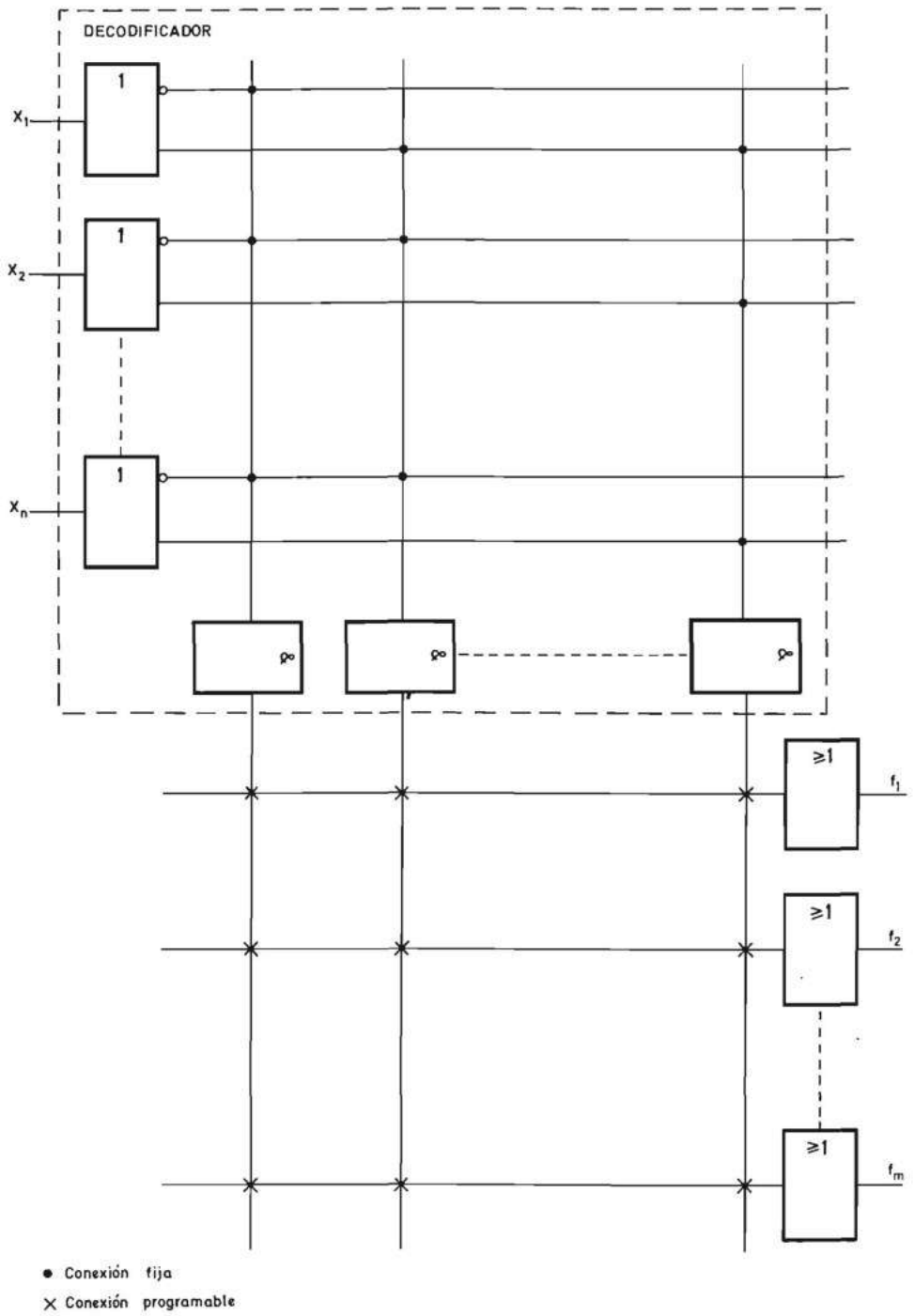


FIGURA 3.114.—**Esquema simplificado de la memoria pasiva de la figura 3.113.**

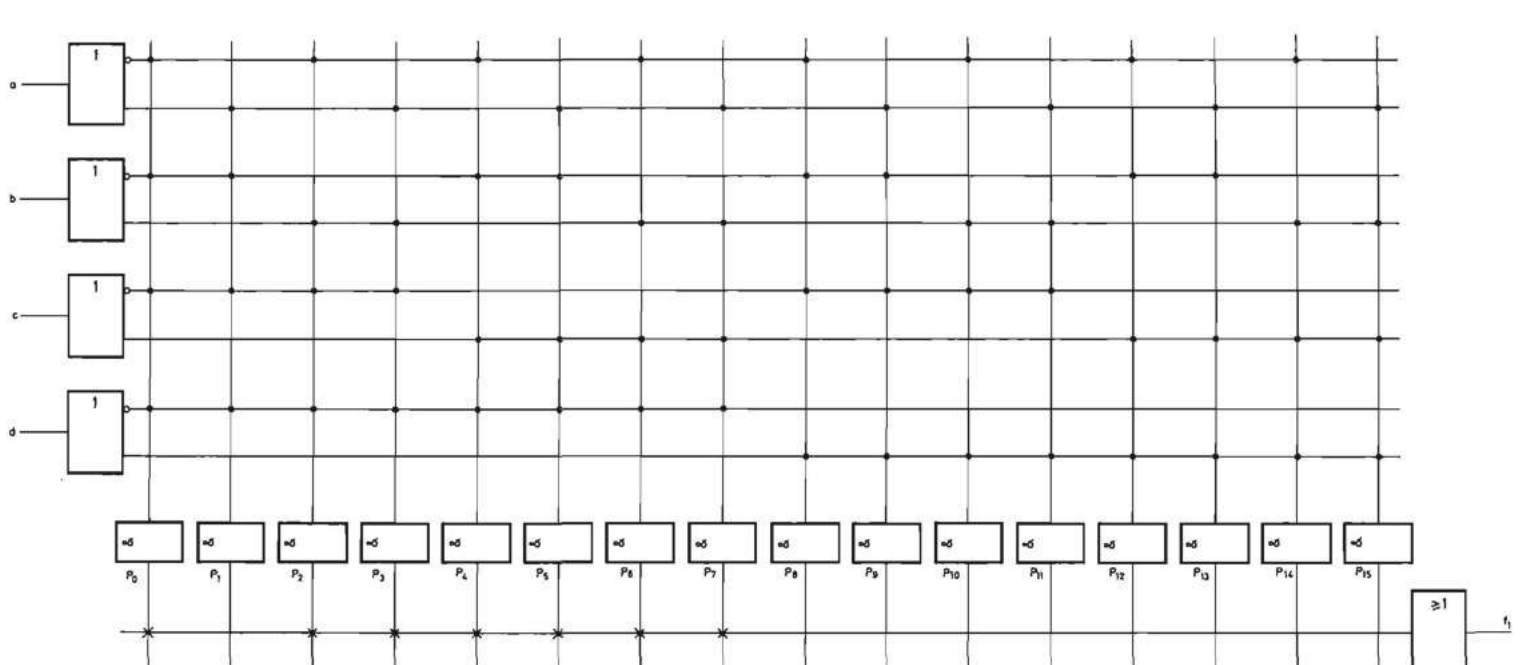


FIGURA 3.115.—**Esquema de la programación de una memoria pasiva de cuatro variables de entrada para realizar las funciones f_1 y f_2 de la tabla 3.32.**

Las memorias pasivas son útiles fundamentalmente para la construcción de tablas, generadores de caracteres, convertidores de código y, en general, cualquier aplicación que requiera programar el valor de las variables de salida para todas y cada una de las combinaciones de entrada. El ejemplo más significativo es la memoria que contiene el programa de un microprocesador en aplicaciones de control. La existencia de muchas aplicaciones en las que no es necesario programar el estado de las salidas para todas las combinaciones de entrada, en especial cuando el número de entradas es elevado, hace preferible en estos casos la utilización de sistemas combinacionales programables incompletos que se estudian en el apartado siguiente.

En el capítulo 7 se estudian con mayor profundidad las memorias pasivas y, en particular, las formas de realizarlas físicamente.

3.8.2.2 Sistemas combinacionales universales programables incompletos (SCUPI). En los sistemas combinacionales programables universales completos que se estudian en el apartado anterior es posible programar de forma independiente el valor de las variables de salida correspondiente a cada una de las combinaciones posibles de las variables de entrada. Pero en la práctica se suelen presentar las siguientes situaciones:

a) Las funciones lógicas sólo toman el valor uno para un cierto número de las combinaciones de las variables de entrada inferior al total 2^n.
b) La expresión canónica de suma de productos se puede simplificar por los métodos numérico o tabular de Karnaugh.
c) La función no está definida para algunas combinaciones de las variables de entrada.

Ninguno de estos tres casos se simplifica cuando la función se realiza mediante

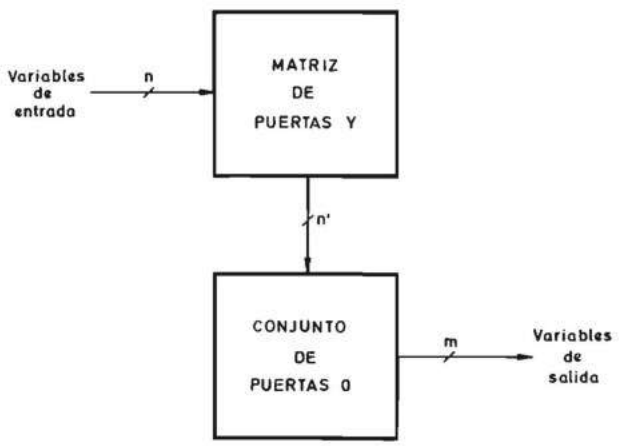


FIGURA 3.116.—**Diagrama de bloques de un sistema combinacional universal programable incompleto realizado mediante una matriz de puertas Y y un conjunto de puertas O.**

un sistema combinacional universal programable completo y de ahí el interés de la utilización de los sistemas combinacionales universales programables incompletos (SCUPI) que están formados por dos o una matriz de puertas lógicas.

Estos sistemas se pueden realizar de dos formas:

a) Mediante una matriz de puertas Y conectada a un conjunto de puertas O (figura 3.116). Según la forma en que se realiza la conexión de la matriz de puertas Y con el conjunto de puertas O, se tienen dos tipos diferentes de SCUPI:

—Las matrices lógicas programables [en inglés «Programmable Logic Array» (PLA)]

—Las matrices lógicas Y-programables [en inglés «Programmable Array Logic» (PAL)]

b) Mediante una matriz de puertas universales.

En sucesivos apartados se estudian los diferentes tipos de SCUPI.

3.8.2.2.1 Matrices lógicas programables [Programmable logic array (PLA)]. Las matrices lógicas programables están constituidas por una matriz programable de n' ($n' < 2^n$, siendo n el número de variables de entrada) puertas Y y una matriz programable de m puertas O (OR). Las puertas Y poseen n entradas que se unen a cada variable de entrada a través de una conexión que puede ser eliminada (figura 3.117).

El número n' es menor que 2^n precisamente porque en la práctica nunca es necesario realizar todos los productos canónicos posibles.

Para programar la matriz lógica programable es necesario realizar las siguientes acciones:

— Suprimir las conexiones adecuadas de la matriz de puertas Y para que la salida de cada una de ellas represente un determinado producto lógico.
— Suprimir las conexiones adecuadas de la matriz de puertas O para que cada variable de salida sea la suma lógica de las salidas de las puertas Y convenientes.

El esquema de la figura 3.117 se puede representar de forma simplificada tal como se indica en la figura 3.118.

Para simplificar el esquema de un sistema digital que posea una matriz lógica programable como las representadas en las figuras 3.117 y 3.118, se puede utilizar el diagrama de bloques de la figura 3.119 en el que se indican por separado la matriz de n' puertas Y y la matriz de m puertas O.

En este diagrama se utilizan los nuevos símbolos normalizados colocando debajo de los indicativos de las puertas Y (&) y O (≥ 1) la cantidad de ellas que es respectivamente n' y m, y el número de entradas de cada una que es respectivamente $2n$ y n'.

Un diagrama de bloques aún más sencillo que constituye un símbolo lógico, es el de la figura 3.120 en el que se utiliza un único símbolo lógico al que se asignan las siglas PLA junto con el número n' de puertas Y que forman parte de la matriz.

A continuación se expone mediante un ejemplo la forma de realizar una función con una matriz lógica programable.

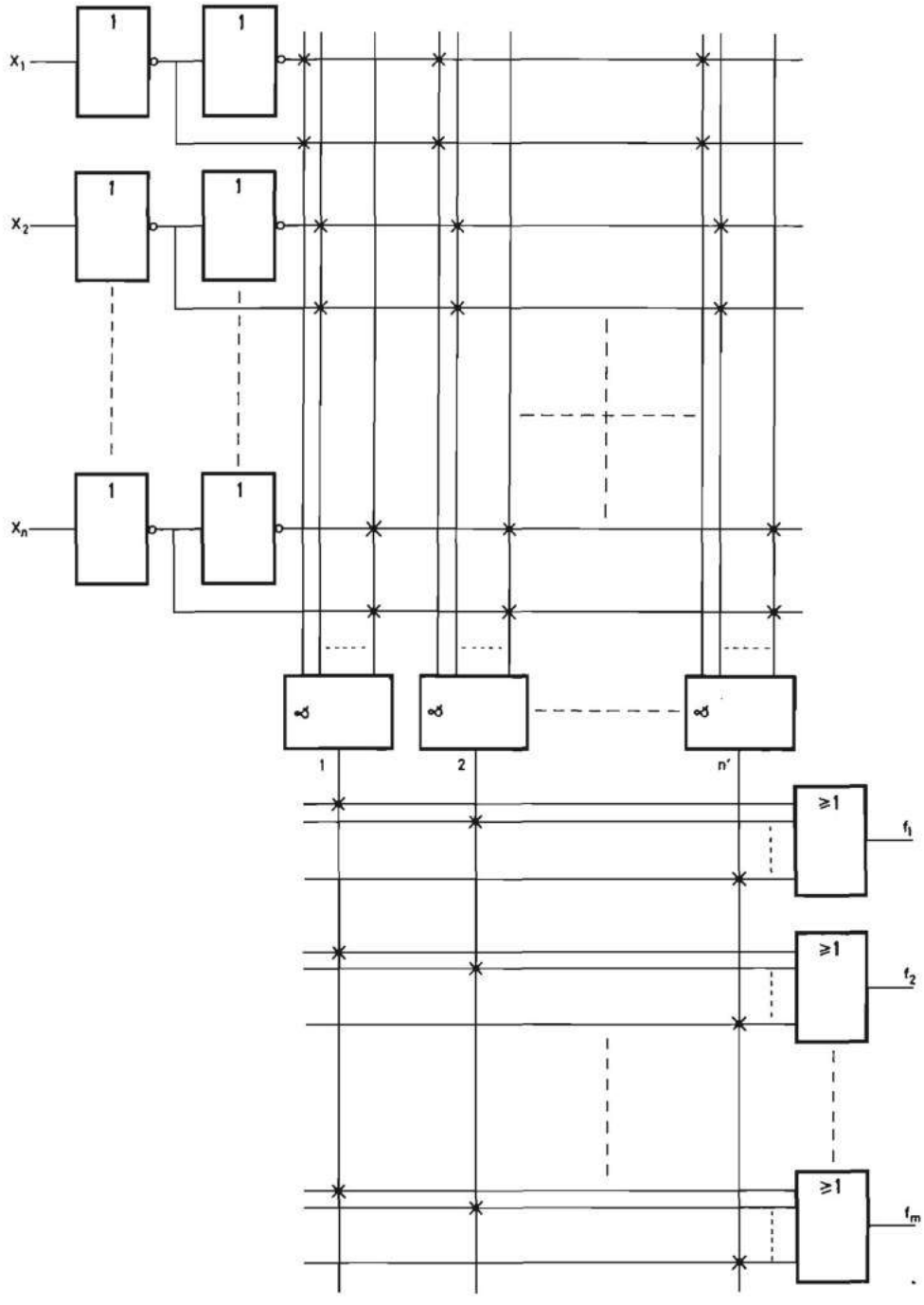


FIGURA 3.117.—**Matriz lógica programable (PLA) de *n* variables de entrada, *n* puertas Y y *m* variables de salida.**

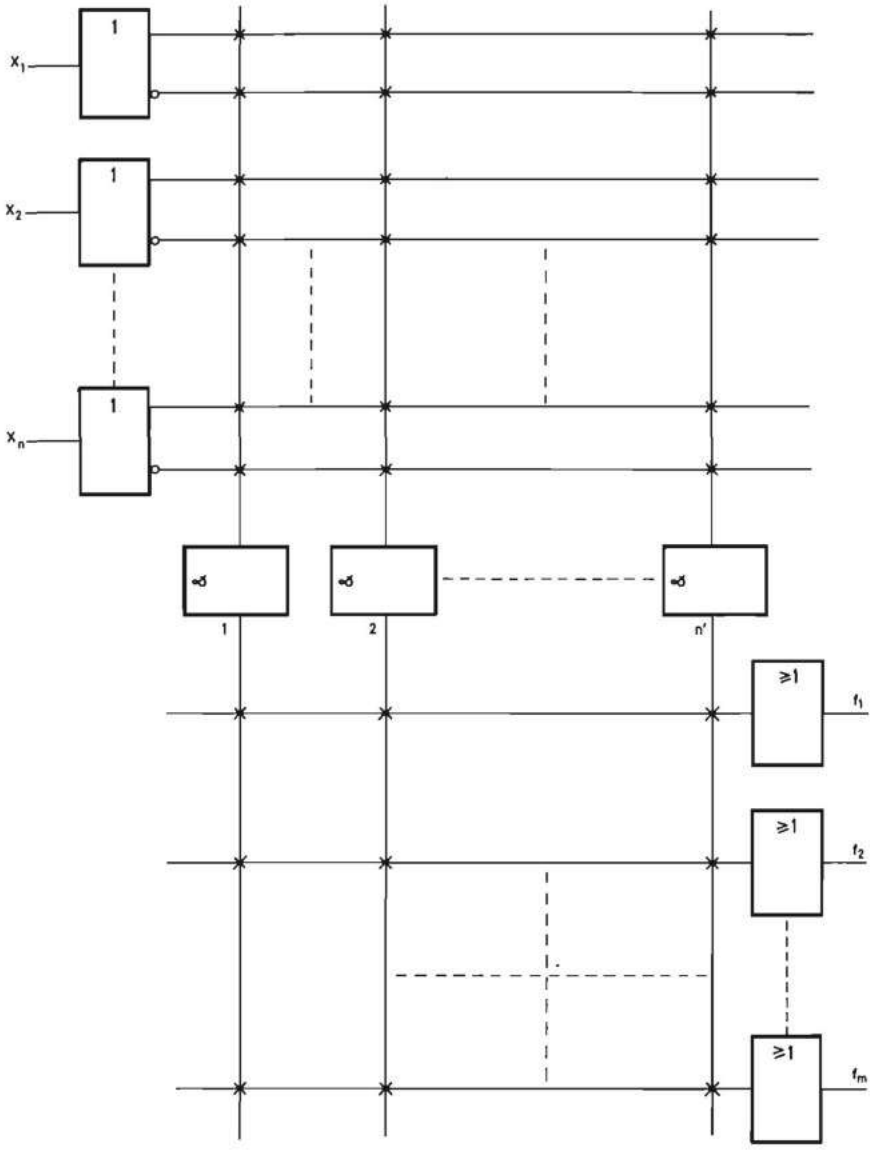


FIGURA 3.118.—**Esquema simplificado de la matriz lógica programable de la figura 3.117.**

Ejemplo 3.11: Realizar mediante una matriz lógica programable las funciones f_1 y f_2 cuya tabla de verdad se representa en la tabla 3.32.

El número de variables de entrada es 4 y el de variables de salida 2 y, por lo tanto, la matriz lógica utilizada ha de tener esta capacidad mínima.

En primer lugar, a partir de la tabla 3.32 se deducen las expresiones canónicas de sumas de productos de las dos funciones:

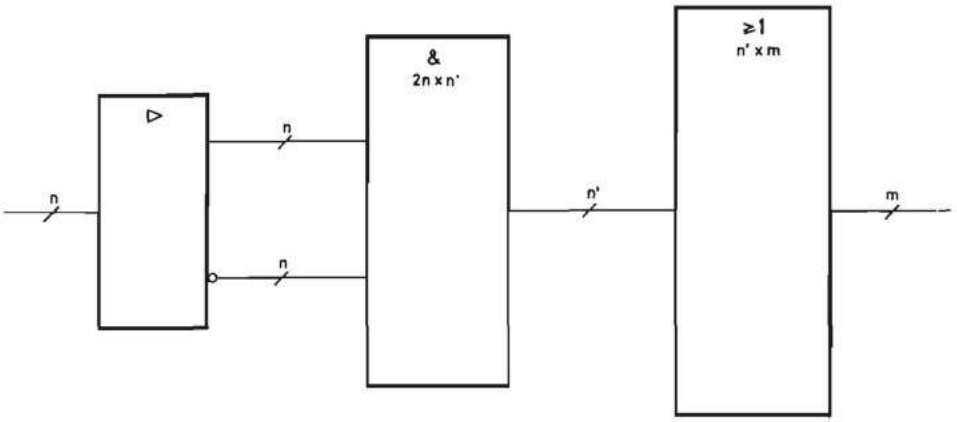


FIGURA 3.119.—**Diagrama de bloques de una matriz lógica programable.**

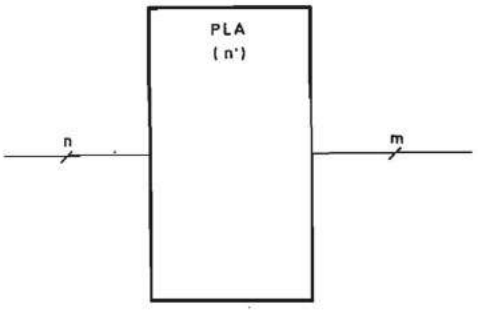


FIGURA 3.120.—**Símbolo lógico de una matriz lógica programable.**

$$f_1 = \sum_4 (0, 2, 3, 4, 5, 6, 7)$$
$$f_2 = \sum_4 (0, 2, 3, 6, 8, 10, 14)$$

El número de productos canónicos diferentes de ambas funciones es 10.

Es posible realizar directamente ambas expresiones con una matriz lógica programable.

La matriz ha de tener un número mínimo de productos igual al de productos canónicos diferentes de ambas funciones, que es 10. En la figura 3.121 se representa su esquema sin programar.

Para realizar la programación se deduce la expresión algebraica de cada producto canónico:

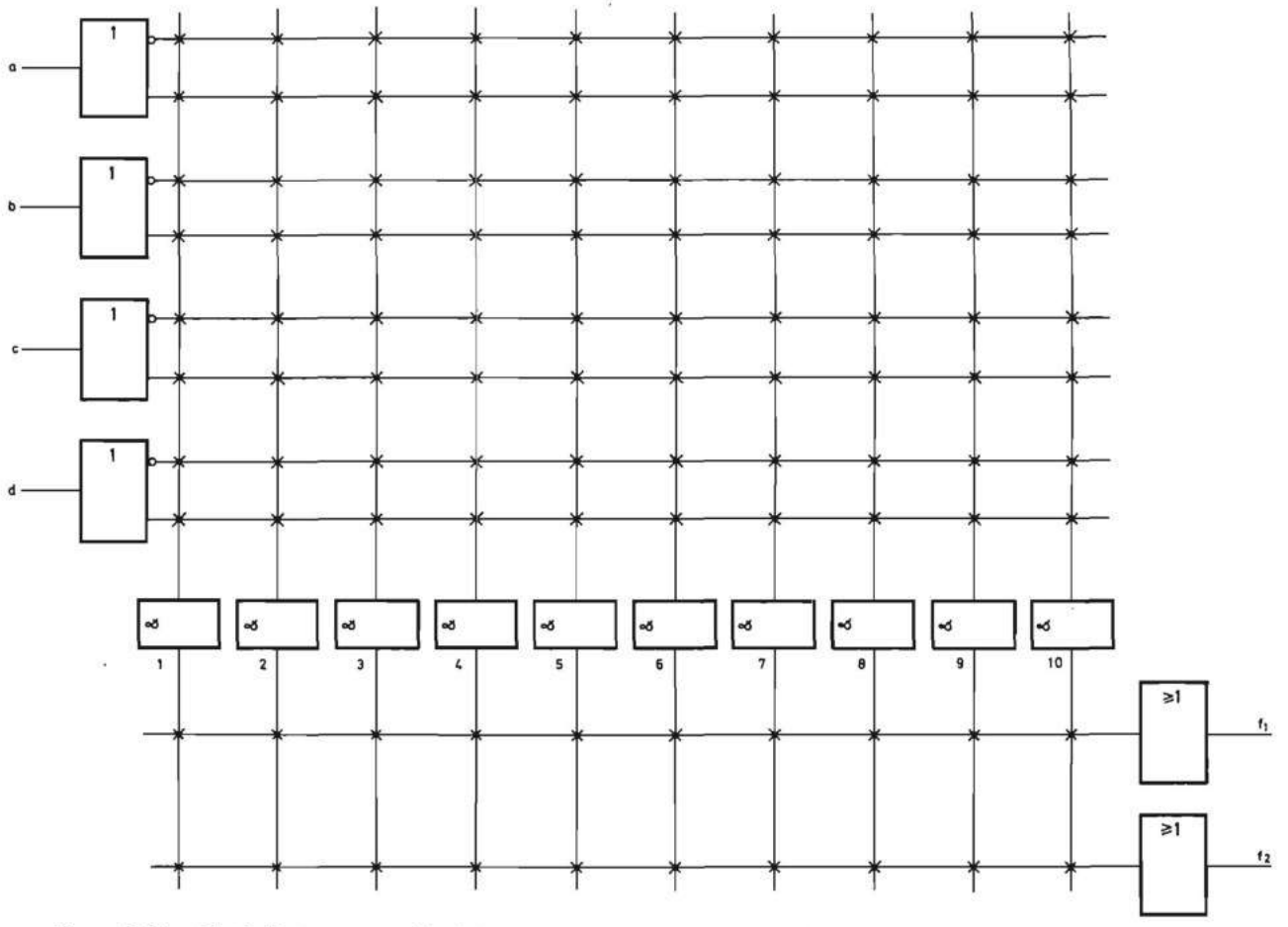


FIGURA 3.121.—**Matriz lógica programable de 4 variables de entrada, 10 productos lógicos y dos variables de salida, sin programar.**

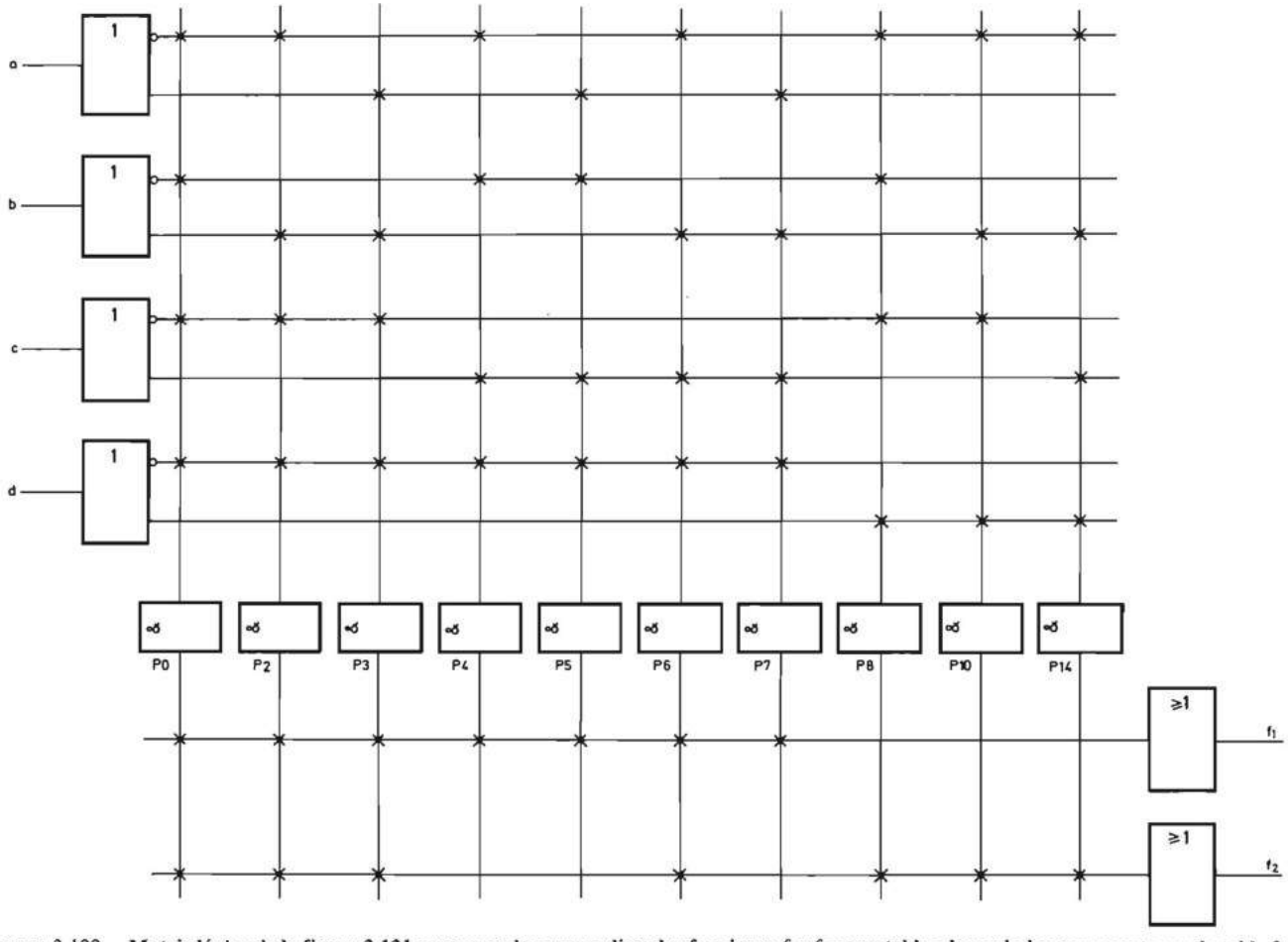


FIGURA 3.122.—**Matriz lógica de la figura 3.121 programada para realizar las funciones f_1 y f_2 cuyas tablas de verdad se representan en la tabla 3.32.**

$$
\begin{aligned}
P_0 &= \bar{a}\,\bar{b}\,\bar{c}\,\bar{d} \\
P_2 &= \bar{a}\,b\,\bar{c}\,\bar{d} \\
P_3 &= a\,b\,\bar{c}\,\bar{d} \\
P_4 &= \bar{a}\,\bar{b}\,c\,\bar{d} \\
P_5 &= a\,\bar{b}\,c\,\bar{d} \\
P_6 &= \bar{a}\,b\,c\,\bar{d} \\
P_7 &= a\,b\,c\,\bar{d} \\
P_8 &= \bar{a}\,\bar{b}\,\bar{c}\,d \\
P_{10} &= \bar{a}\,b\,\bar{c}\,d \\
P_{14} &= \bar{a}\,b\,c\,d
\end{aligned}
$$

En la figura 3.122 se representa la matriz lógica programable una vez programada para realizar las funciones f_1 y f_2. Cada puerta Y realiza uno de los productos lógicos que forman parte de una o ambas funciones f_1 y f_2. Por ejemplo, en la puerta Y que realiza el producto P_3 se suprimen las conexiones de sus entradas a las variables $\bar{a}$, $\bar{b}$, c y d y se mantienen las conexiones a las variables a, b, $\bar{c}$ y $\bar{d}$. Las entradas de las puertas O correspondientes a los productos que no forman parte de la función se suprimen también. Por ejemplo, en la puerta O que realiza f_1 se suprimen las uniones de las entradas correspondientes a P_8, P_{10} y P_{14}.

Pero la simplificación de las expresiones canónicas de suma de productos de f_1 y f_2 permite reducir las dimensiones de la matriz lógica programable necesaria. Dicha simplificación se puede realizar, considerándolas en conjunto como una multifunción, por el método tabular de Karnaugh descrito en el apartado 3.4.

En la figura 3.123 se representan las tablas de Karnaugh de suma de productos a partir de las cuales se deducen las expresiones:

$$
\begin{aligned}
f_1 &= \bar{a}\,\bar{d} + c\,\bar{d} + b\,\bar{c}\,\bar{d} \\
f_2 &= \bar{a}\,\bar{c} + \bar{a}\,b + b\,\bar{c}\,\bar{d}
\end{aligned}
$$

Partiendo de estas expresiones se deduce que la matriz lógica programable necesaria ha de tener solamente cinco puertas Y correspondientes a los cinco productos diferentes existentes en las expresiones de f_1 y f_2. El lector puede obtener fácilmente la matriz programada representada en la figura 3.124. En las puertas Y que realizan un producto lógico en el cual falta alguna variable, se suprimen las uniones correspondientes a dicha variable y su inversa.

Las matrices lógicas programables son los SCUPI más flexibles, porque en ellos es posible programar la conexión de cada producto a todas y cada una de las puertas O de salida, pero en contrapartida presentan la característica de que es necesario programar dos matrices.

3.8.2.2.2 Matrices lógicas Y-programables (PAL). El diagrama de bloques de las matrices lógicas Y-programables también coincide con el de la figura 3.115 pero se diferencian de las matrices lógicas programables en que las entradas de las puertas O están conectadas rígidamente a un determinado número de puertas Y. En general, si la PAL posee n' productos y m salidas, cada puerta O se conecta a n'/m productos diferentes. En la figura 3.125 se representa como ejemplo una

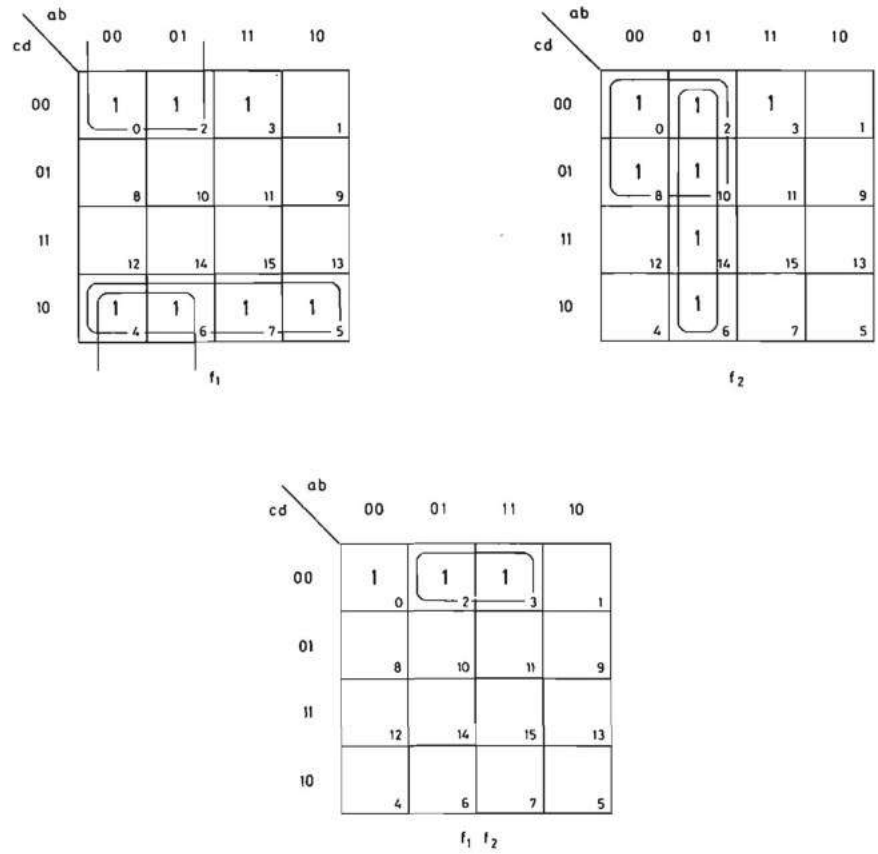


FIGURA 3.123.—**Tablas de Karnaugh de las funciones f_1 y f_2 cuyas tablas de verdad se representan en la tabla 3.32.**

PAL de doce productos lógicos y tres puertas O, conectadas a cuatro productos cada uno.

El nombre de PAL fue registrado por el primer fabricante que se decidió a integrarlas en un único circuito integrado (la empresa ya desaparecida «Monolithic Memories»). Su elección fue debida a que al estar registrado el nombre de PLA, al citado fabricante no se le ocurrió otra cosa que permutar la L por la A. Por ello el nombre de PAL es un ejemplo, cada vez más frecuente en microelectrónica, de la importancia de la imagen de marca en las denominaciones de los circuitos.

Las PAL son menos flexibles que las PLA, y necesitan más puertas Y que éstas porque si un producto lógico ha de formar parte de dos salidas, ha de ser programado dos veces. En compensación, las PAL poseen un menor tiempo de propagación, menor disipación y, lo que es más importante, ocupan una menor superficie de silicio, y su programación es evidentemente más sencilla.

En la figura 3.126 se representa el diagrama de bloques de una matriz lógica Y-

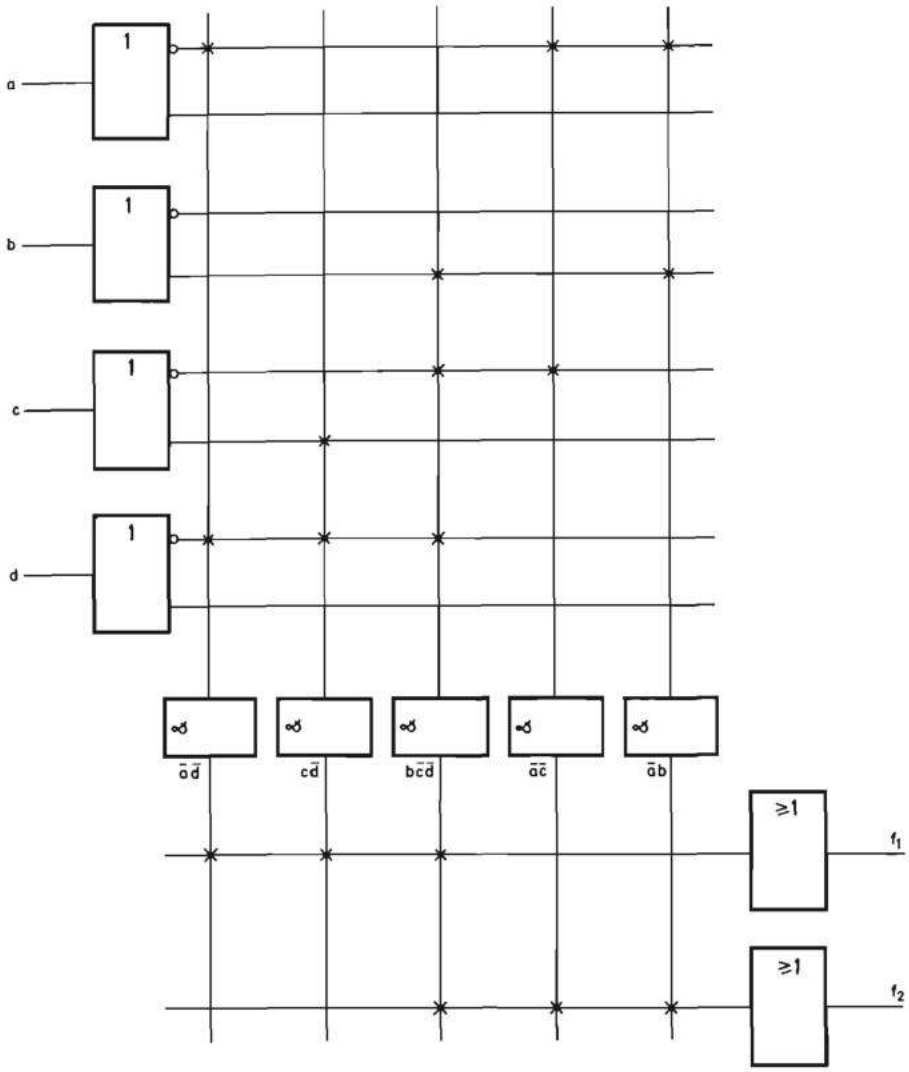


FIGURA 3.124.—**Matriz lógica programable programada para realizar las expresiones mínimas de suma de productos de f_1 y f_2.**

programable (PAL) de *n'* productos y *m* puertas O con *n'/m* entradas cada una y en la figura 3.127 el diagrama de bloques simplificado que constituye un símbolo lógico.

A continuación se expone mediante un ejemplo la forma de realizar una función lógica con una matriz lógica Y-programable.

Ejemplo 3.12: Realizar mediante una matriz lógica Y-programable las mismas funciones realizadas en el ejemplo 3.11 con una matriz lógica programable (PLA). Dichas funciones tienen las expresiones canónicas:

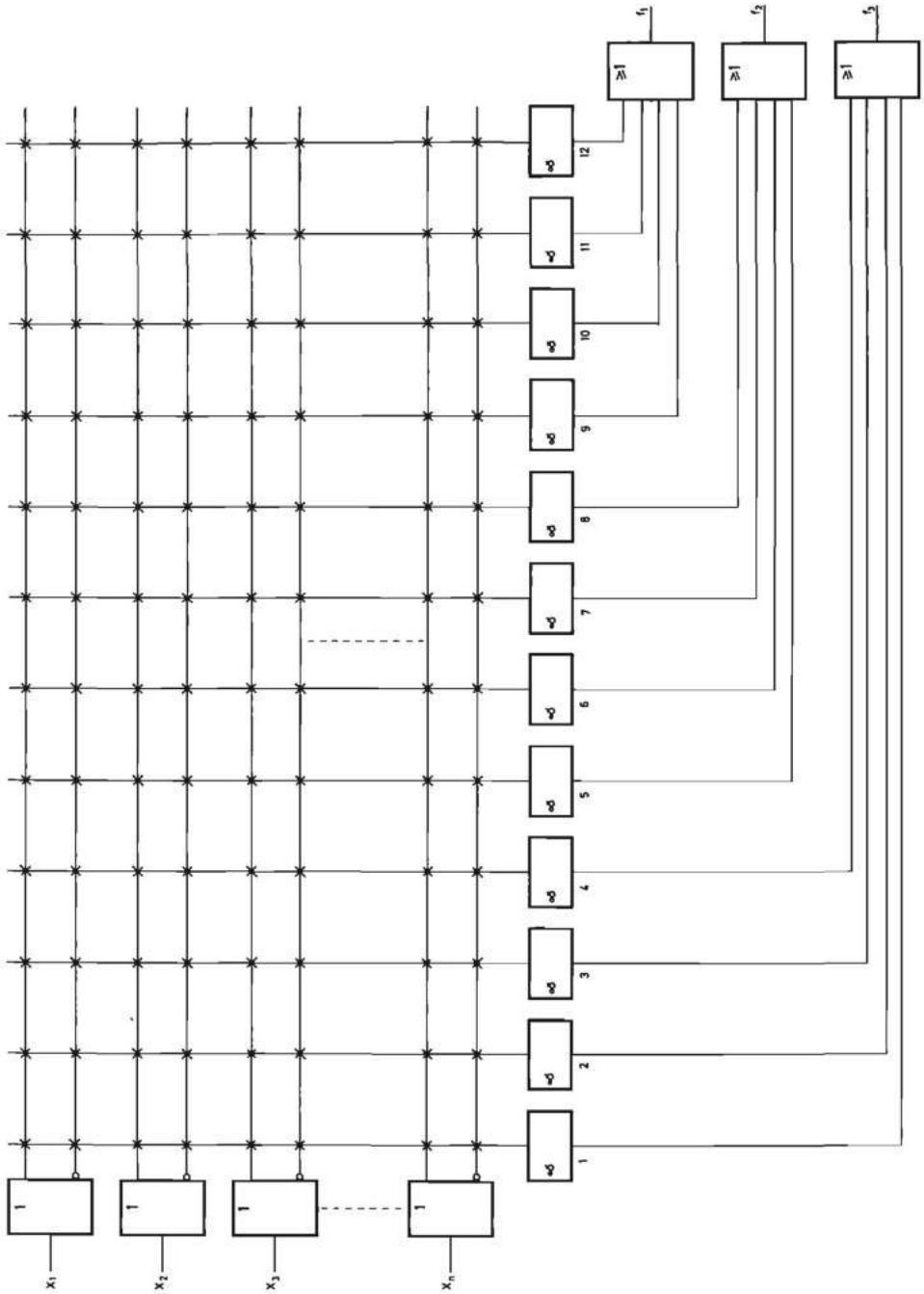


FIGURA 3.125.—**Matriz lógica Y-programable (PAL) de *n* variables de entrada, 12 productos y 3 variables de salida.**

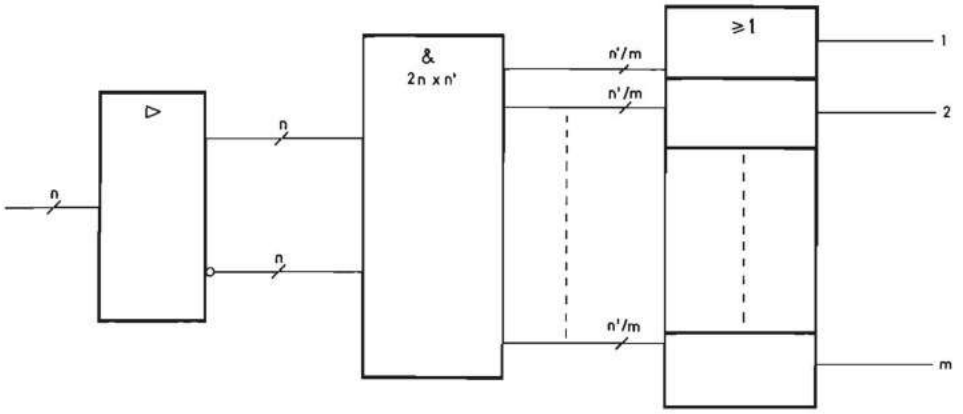


FIGURA 3.126.—**Diagrama de bloques de una matriz lógica Y-programable (PAL).**

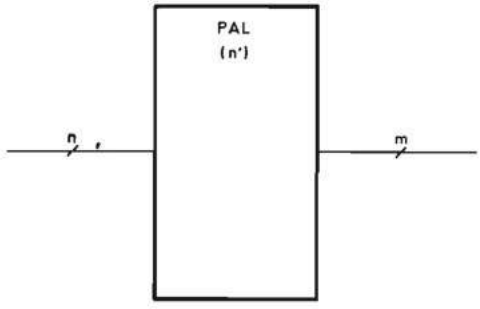


FIGURA 3.127.—**Símbolo lógico de una matriz lógica Y-programable (PAL).**

$$f_1 = \sum_4 (0,2,3,4,5,6,7)$$
$$f_2 = \sum_4 (0,2,3,6,8,10,14)$$

Dado que cada expresión tiene 7 productos, para realizarlas directamente con una PAL, ésta ha de tener como mínimo 14 puertas Y y 2 puertas O, de acuerdo con el esquema representado sin programar en la figura 3.128.

En la figura 3.129 se representa la PAL una vez programada para realizar las funciones f_1 y f_2. Se observa que existen dos puertas Y que realizan el producto P0 y otro tanto sucede con P2, P3 y P6.

La simplificación de las expresiones canónicas de suma de productos permite también reducir la complejidad de la PAL mínima necesaria.

Las expresiones mínimas de productos de sumas de f_1 y f_2, obtenidas a partir de las tablas de Karnaugh de la figura 3.123, son:

$$f_1 = \bar{a}\bar{d} + c\bar{d} + b\bar{c}\bar{d}$$
$$f_2 = \bar{a}\bar{c} + \bar{a}b + b\bar{c}\bar{d}$$

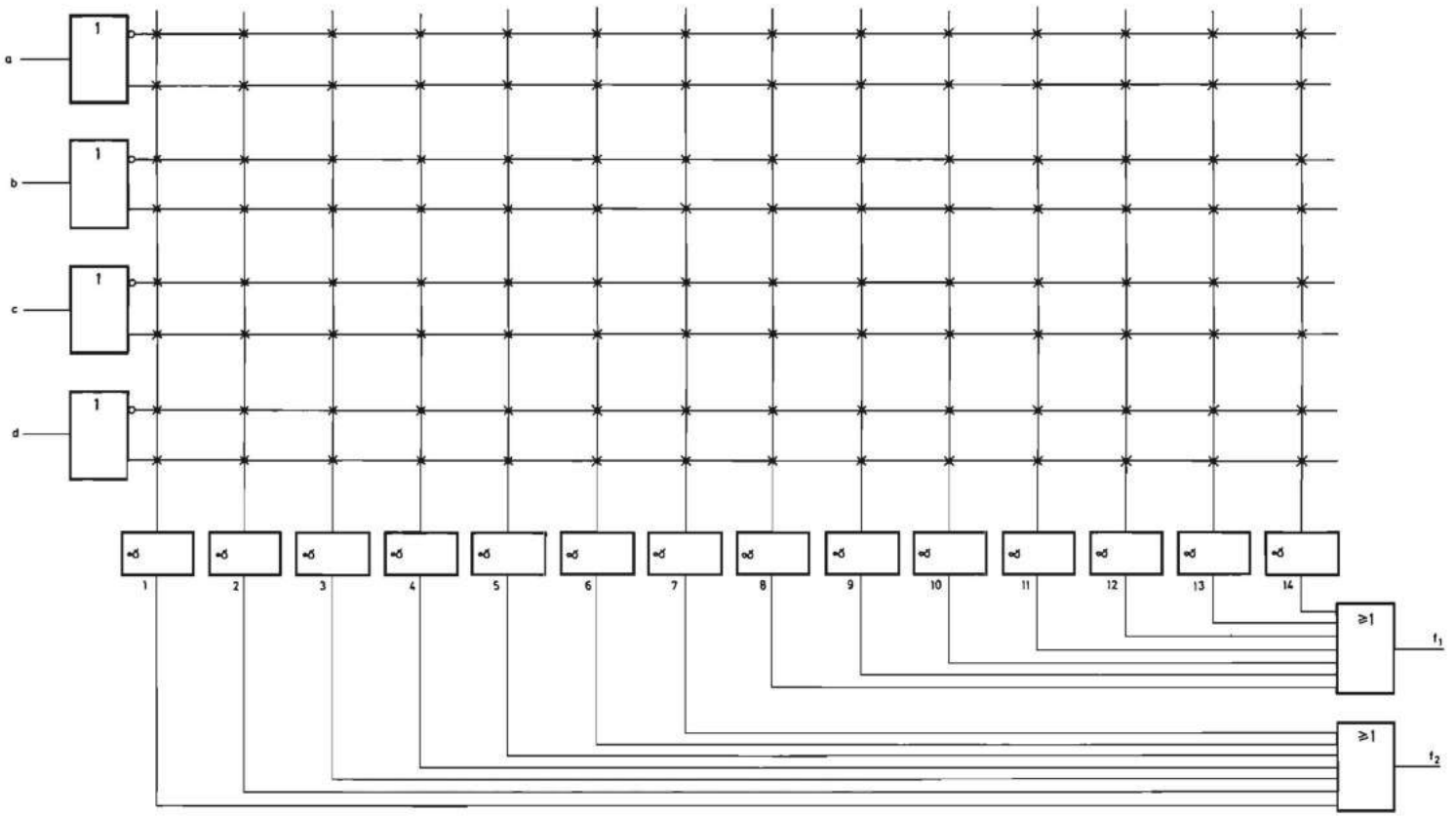


FIGURA 3.128.—**Matriz lógica Y-programable mínima necesaria para realizar las funciones f_1 y f_2 (tabla 3.32) sin programar.**

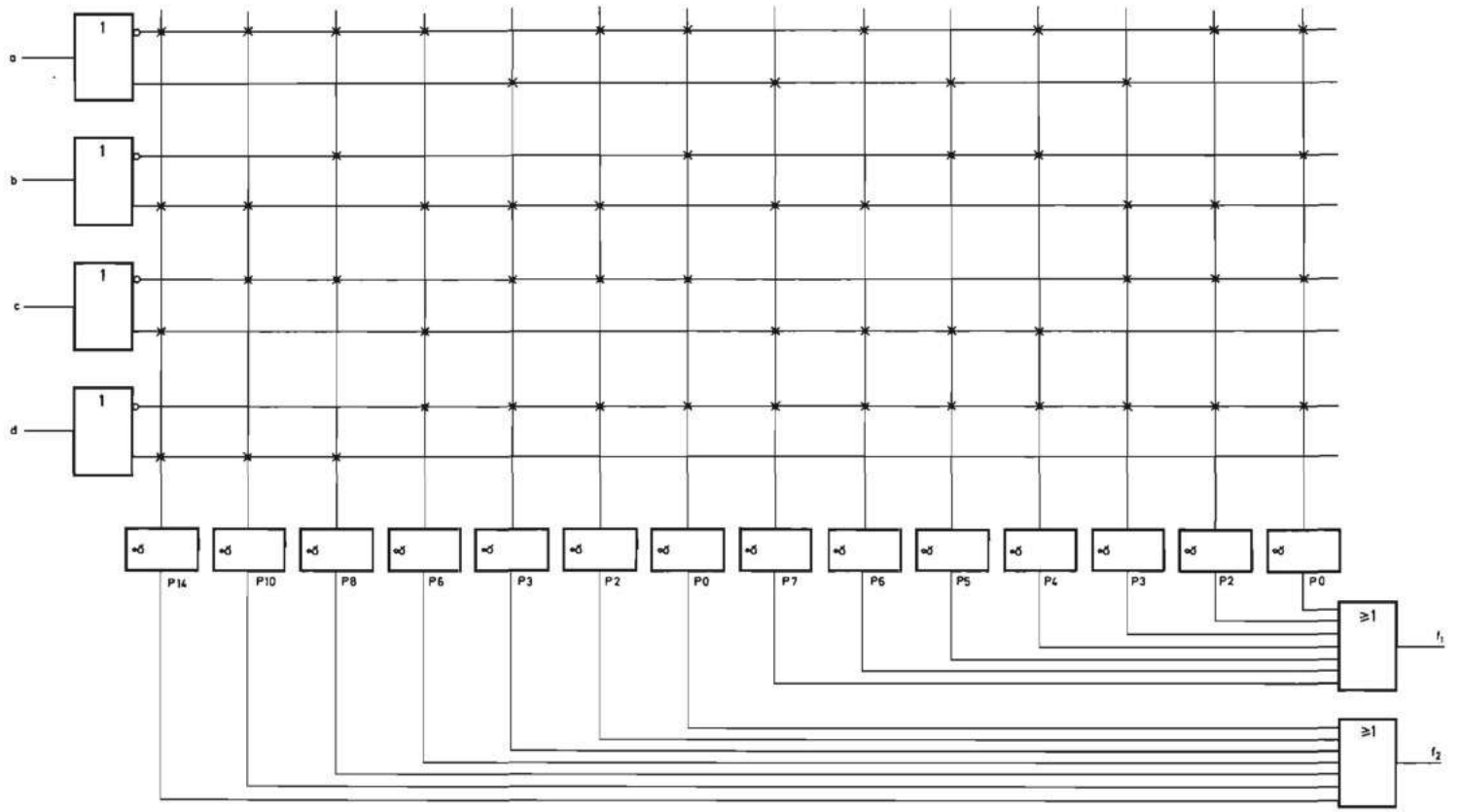


FIGURA 3.129.—**Matriz lógica Y-programable de la figura 3.128 programada para realizar las funciones f_1 y f_2.**

La matriz lógica Y-programable mínima necesaria ha de tener seis puertas Y. En la figura 3.130 se representa su esquema con la programación adecuada cuya comprobación se recomienda al lector.

Las matrices lógicas Y-programables (PAL) adquieren su verdadero interés combinándolas con registros y con un conjunto de recursos lógicos para obtener los denominados dispositivos lógicos programables (DLP) [en inglés «Programmable Logic Devices» (PLD)]. Por ello las PAL que poseen solamente una matriz de puertas Y y un conjunto de puertas O no se suelen realizar, en general, en la actualidad en circuitos independientes.

Los DLP permiten realizar tanto circuitos combinacionales como secuenciales

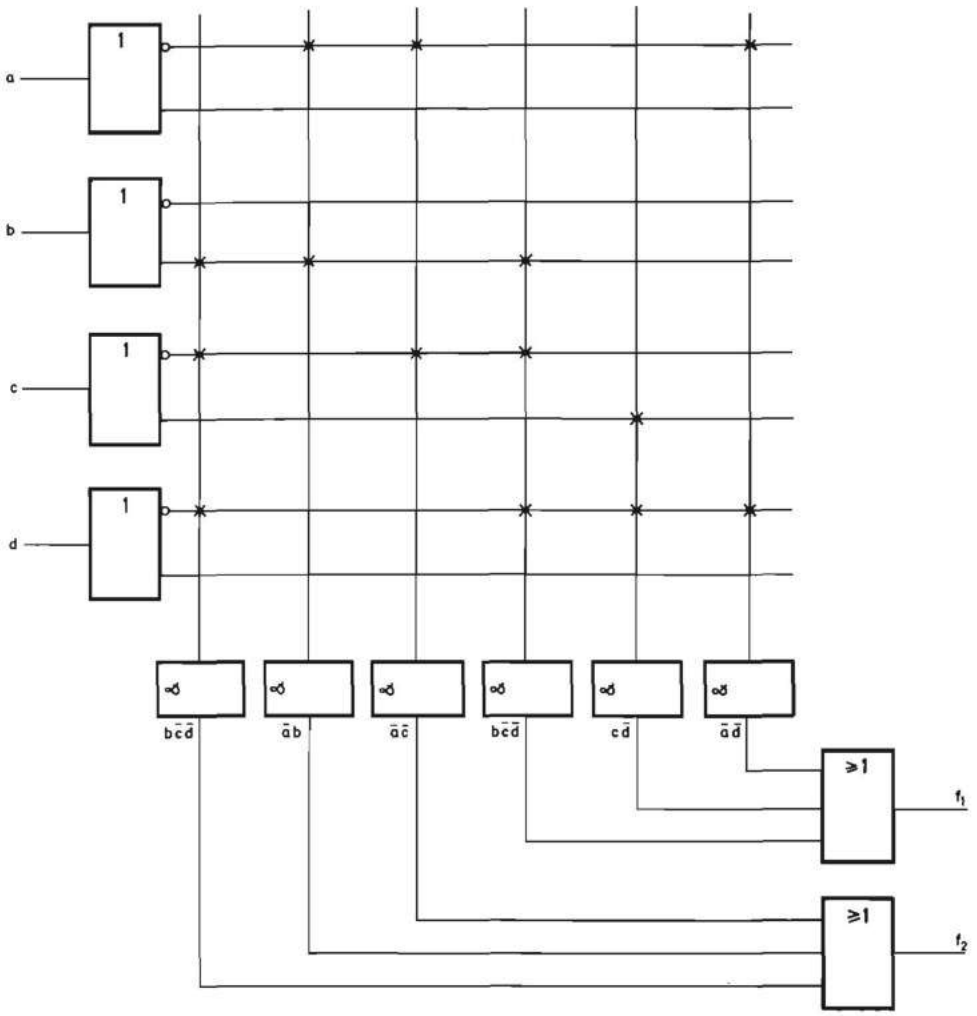


FIGURA 3.130.—**Matriz lógica Y-programable mínima programada para realizar las expresiones simplificadas de suma de productos de f_1 y f_2.**

y por ello su estudio se realiza en el capítulo 6. Los DLP son además circuitos integrados digitales monolíticos configurables y por ello se estudian también en el capítulo 5, en el apartado dedicado a los circuitos integrados digitales monolíticos normalizados. No obstante, un estudio en profundidad de los DLP se sale fuera de los límites de este libro. Al lector interesado se le remite a los libros «Controladores lógicos y autómatas programables» [MAND 92] y «Sistemas digitales configurables y sus aplicaciones. Tomo I: Dispositivos lógicos programables» desarrollados por varios profesores del Departamento de Tecnología Electrónica que, a su vez, son miembros del Instituto de Electrónica Aplicada Pedro Barrié de la Maza de la Universidad de Vigo.

3.8.2.2.3 Ampliación de la capacidad de las matrices lógicas programables (PLA) y las matrices lógicas Y-programables (PAL). Los bloques funcionales poseen un cierto número n de variables de entrada, n' de puertas Y y m de puertas O y variables de salida. En la práctica puede ser necesario realizar una PLA o una PAL de un número n_1 de variables de entrada superior a n, o un número n'_1 de productos superior a n' o un número de variables de salida m_1 superior a m.

Aunque la elevación de la capacidad de integración hace que los métodos de ampliación tengan cada vez menor importancia en la utilización de circuitos comerciales, es interesante conocerlos porque son útiles en las técnicas de diseño microelectrónico.

A continuación se analiza la ampliación de cada uno de los factores.

a) Elevación del número de productos.

Si el número n' de productos lógicos de un bloque funcional no es suficiente, se puede hacer igual a un múltiplo de n', conectando las variables de entrada a varios módulos y uniendo entre sí las salidas correspondientes a las funciones que utilicen productos de varios módulos (fig. 3.131). Para que esta unión sea posible es necesario que las puertas O del bloque funcional tengan una de las siguientes características:

a) Poseer como carga del transistor de salida una resistencia, tal como se describe en el apartado 5.4.4.3.2.
b) Poseer tres estados para lo cual el bloque funcional ha de tener una entrada de desinhibición/inhibición (*EN*) que cuando adopte un cierto estado lógico (por ejemplo el cero lógico), provoque el tercer estado de la salida y en caso contrario aparezca en ella el estado lógico producido por el circuito.

En la figura 3.132 se realiza el acoplamiento de dos matrices de n' productos para obtener una matriz de $2n'$ productos. Las entradas *EN* se conectan entre sí a través de un inversor. La variable X_{n+1} conectada a ellas ha de aparecer en forma directa en los productos de la matriz superior y en forma inversa en los de la matriz inferior. La variable que se conecte a la entrada *EN* debe ser, en general, aquella que haga que los números de productos de ambas matrices sean lo más próximos posibles. Un ejemplo aclarará lo que se acaba de exponer.

Ejemplo 3.13: Sea la función $f = \sum_3 (0, 1, 2, 4, 6)$ (tabla 3.33) que se desea

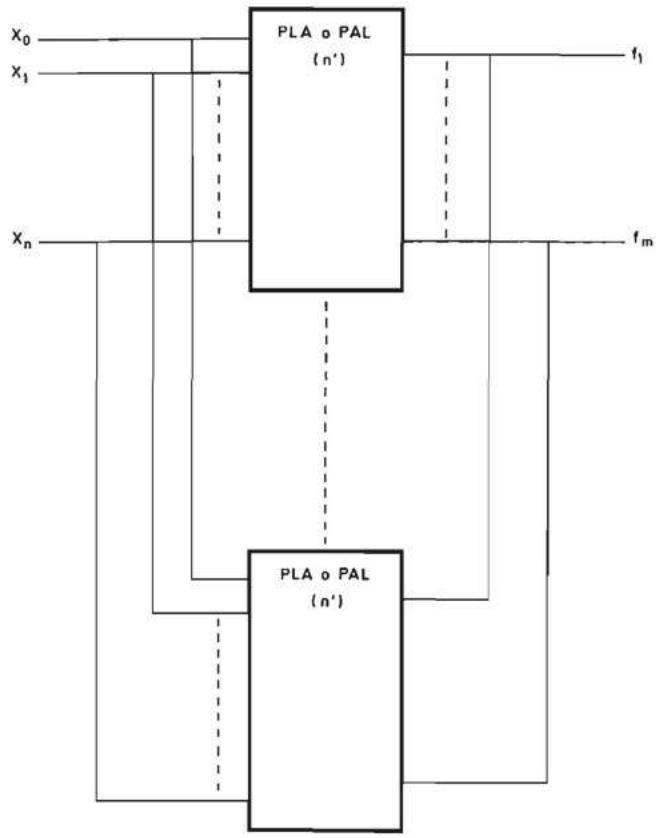


FIGURA 3.131.—**Elevación del número de productos de una PLA o una PAL.**

realizar en forma canónica mediante uña matriz lógica programable de dos variables de entrada y tres productos canónicos, que posee una salida de tres estados y una entrada de desinhibición (*EN*) que se supone que produce la salida de tres estados cuando se encuentra en nivel cero.

Las expresiones algebraicas de los productos canónicos de f son:

$$P_0 = \overline{a}\,\overline{b}\,\overline{c}$$
$$P_1 = a\,\overline{b}\,\overline{c}$$
$$P_2 = \overline{a}\,b\,\overline{c}$$
$$P_4 = \overline{a}\,\overline{b}\,c$$
$$P_6 = \overline{a}\,b\,c$$

Observando estos productos se comprueba que la variable b está en forma directa en dos de ellos y en forma inversa en tres, al igual que la variable c. Cualquiera de ellas puede ser utilizada para gobernar la entrada de inhibición. Por el contrario, la variable a no puede ser utilizada porque se encuentra en forma inver-

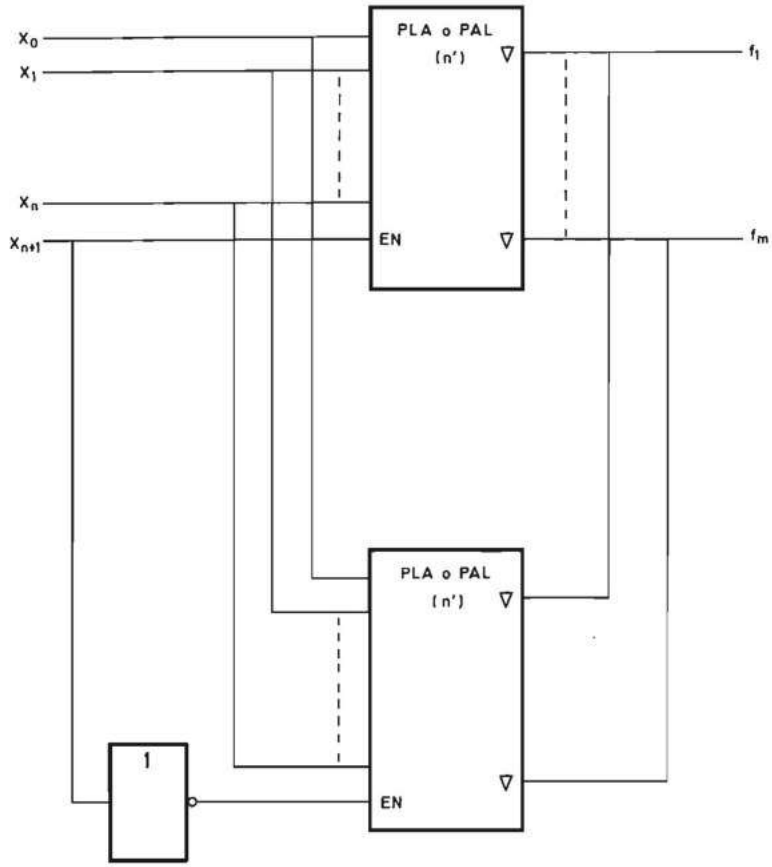


FIGURA 3.132.—**Elevación del número de productos de una PLA o una PAL que posee salida de tres estados.**

c	b	a	f_1
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

TABLA 3.33

sa en cuatro productos canónicos y se necesitaría una matriz lógica programable de esta capacidad.

En la figura 3.133 se indica el diagrama de bloques de la solución en la que se controla la entrada de inhibición con la variable *b*. La entrada de desinhibición *EN* de la matriz lógica programable *N*1 se une a la variable *b* y en ella se realizan los productos canónicos P_2 y P_6 y la entrada de desinhibición de la matriz lógica programada *N*2 se conecta a la variable *b* y en ella se realizan los productos canónicos P_0, P_1 y P_4.

b) Elevación del número de variables de salida.

Si el número *m* de variables de salida de un bloque funcional no es suficiente, se eleva el número de salidas acoplando las variables de entrada a los módulos ne-

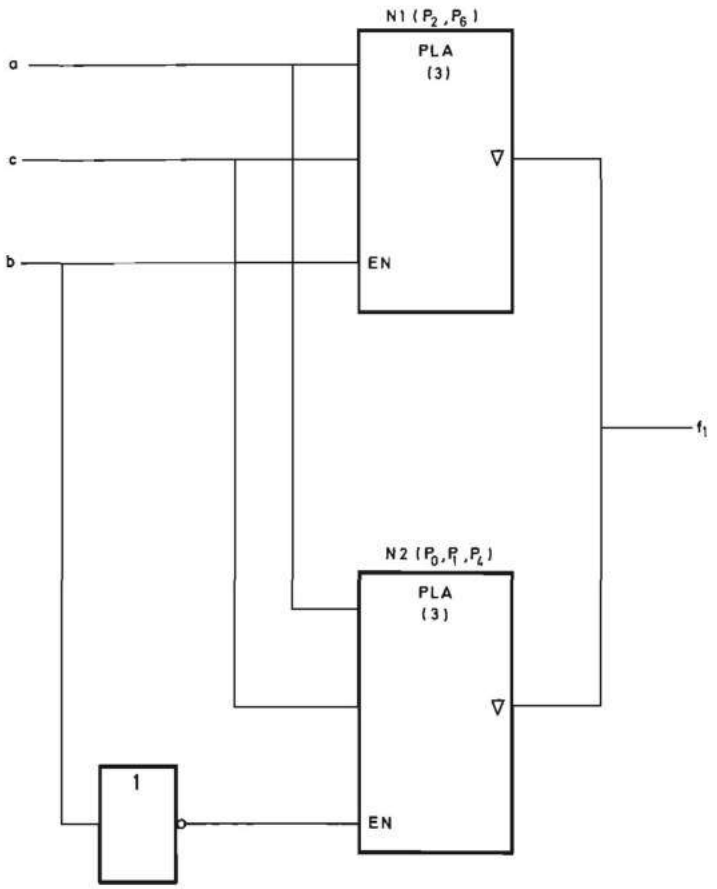


FIGURA 3.133.—**Realización de la función f_1 (tabla 3.33) mediante dos matrices lógicas programables de dos variables de entrada y salida de tres estados con entrada de desinhibición (*EN*).**

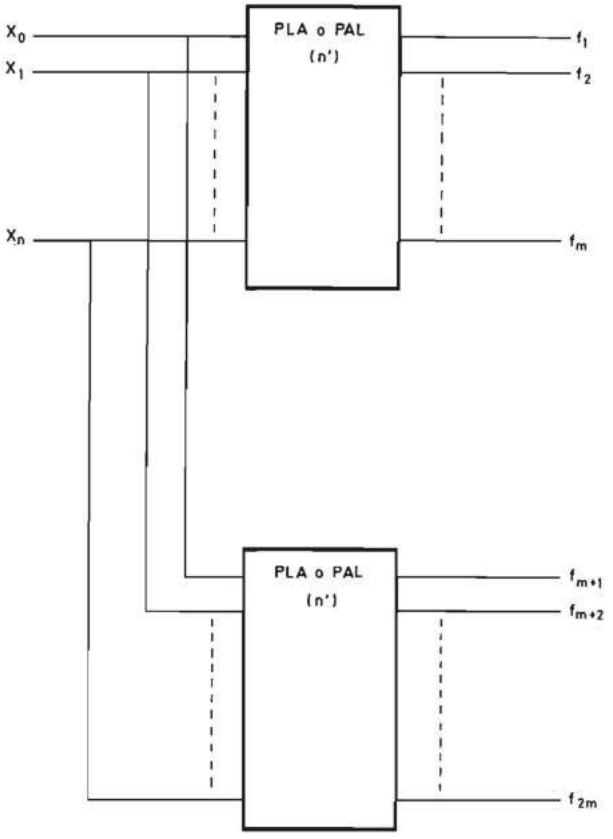


FIGURA 3.134.—**Elevación del número de variables de salida de una PLA o PAL.**

cesarios. En la figura 3.134 se supone que el número de variables de salida está comprendido entre m y $2m$.

c) Elevación del número de variables de entrada.

Si el número n de variables de entrada de un módulo funcional es insuficiente, se puede elevar mediante la entrada de inhibición. En el apartado *a*) se vio la forma de elevar el número de productos mediante dicha entrada y se observó que el número de variables de entrada posibles se eleva en una unidad. La utilización de un decodificador en combinación con las matrices lógicas programables (PLA o PAL) tal como se indica en la figura 3.135 permite elevar el número de variables de entrada. El lector no debe tener ningún problema para comprender su funcionamiento.

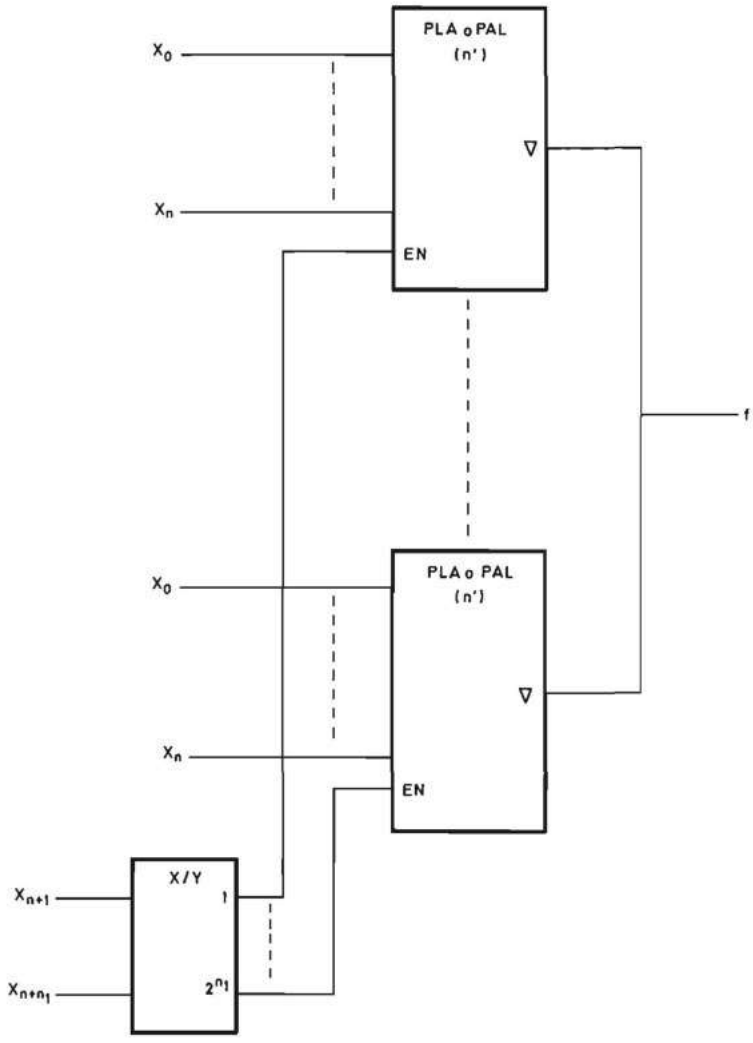


FIGURA 3.135.—**Elevación del número de variables de entrada de una PLA o PAL.**

3.8.2.2.4 Matrices lógicas de puertas universales. Las puertas NO-Y (NAND) y NO-O (NOR) son puertas universales porque mediante una combinación de cualquiera de ambas se puede realizar cualquier función lógica, tal como se indica en el apartado 3.5.1. Por ello, combinándolas con una matriz de conexiones programables se obtiene un sistema combinacional universal programable incompleto (SCUPI).

En la figura 3.136 se representa el esquema básico de un SCUPI formado por un conjunto de puertas NO-Y y una matriz de conexiones programables a cuyas filas

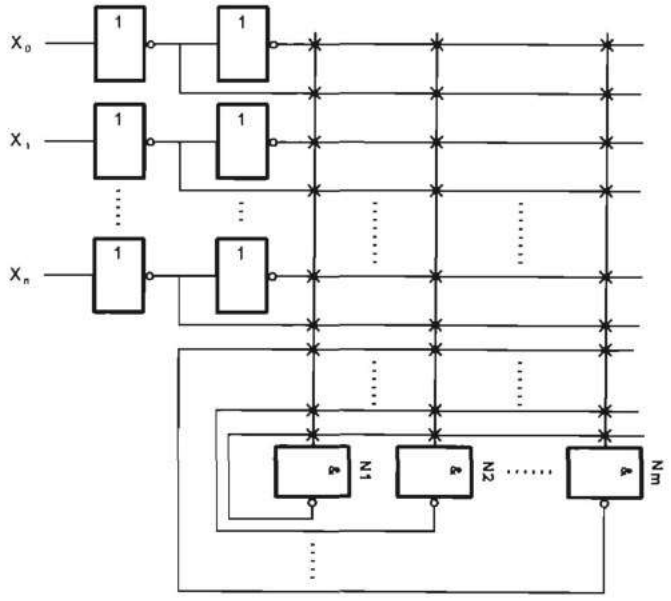


FIGURA 3.136.—**Matriz lógica programable de puertas NO-Y (NAND) realimentadas.**

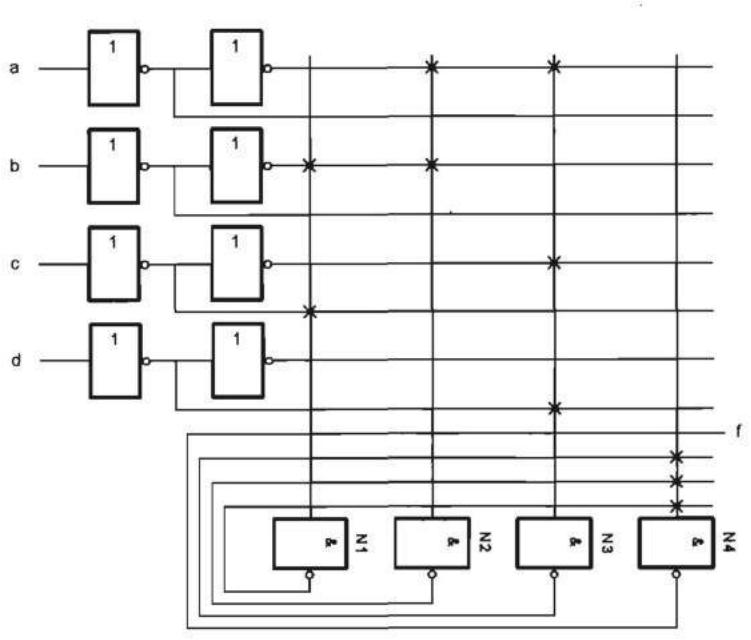


FIGURA 3.137.—**Generación de la función $f = b\bar{c} + ab + ac\bar{d}$ con una matriz lógica programable de puertas NO-Y (NAND) realimentadas.**

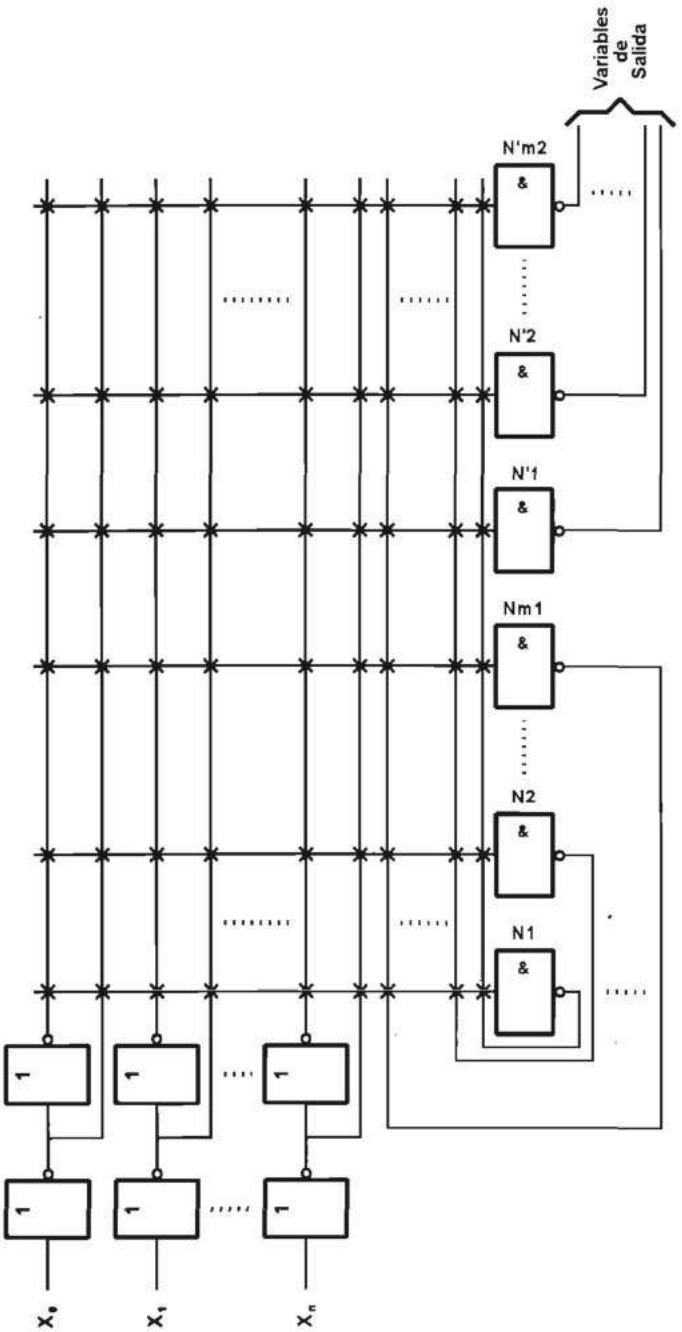


FIGURA 3.138.—**Esquema de una matriz lógica con puertas NO-Y (NAND) realimentadas que posee una matriz de puertas de salida.**

se conectan las variables de entrada (directas o invertidas) y las salidas de las puertas NO-Y.

Mediante la supresión de las conexiones adecuadas se puede obtener cualquier función lógica. En la figura 3.137 se representa una matriz de puertas NO-Y realimentadas que realiza la función $f = b\bar{c}+ab+ac\bar{d}$, que transformada adecuadamente se convierte en:

$$f = \overline{\overline{b\bar{c}}\;\overline{ab}\;\overline{ac\bar{d}}}$$

Para ello la matriz ha de tener como mínimo cuatro puertas NO-Y, una para realizar cada uno de los tres productos y la cuarta para generar f a partir de las otras.

En la figura 3.137 las puertas N1 a N3 generan los tres productos invertidos y N4 genera la función f.

En el ejemplo de la figura 3.137 se observa que la salida de N4 no se conecta a la entrada de ninguna puerta una vez realizada la programación. Por ello, para no complicar innecesariamente la matriz de conexión, el esquema real de una matriz lógica de puertas NO-Y es el representado en la figura 3.138 en la que hay dos conjuntos de puertas NO-Y:

— Un conjunto de m_1 puertas realimentadas.
— Un conjunto de m_2 puertas no realimentadas.

Las matrices lógicas programables realizadas con puertas NO-Y y NO-O, al igual que las PLA y las PAL se combinan con registros y otros recursos lógicos para obtener dispositivos lógicos programables. Para profundizar en su estudio se remite al lector a la misma bibliografía indicada anteriormente.

BIBLIOGRAFIA

[INTE 89] Programmable logic handbook. Intel Corporation. 1989.
[MAND 92] E. Mandado, J. Marcos, S.A. Pérez. Controladores lógicos y autómatas programables. Capítulo 1: Modularidad de entradas y salidas. 2ª edición. Editorial Marcombo. 1992.
[PHIL 89] Semi-custom Programmable Logic Devices. Philips. 1989.
[TEXA 85] The TTL data book. Volume 2. Fuse-programmable identity comparators. SN74ALS526. Texas Instruments. 1984.
[CYPR 89] CMOS, BICMOS Data book. Cypress Semiconductor 1989.

PROBLEMAS

1. En un registro de cuatro bits cuyas salidas están disponibles al exterior se almacena información en el código BCD Aiken.
   a) Realizar la tabla de verdad de un circuito que detecte que el número contenido en el registro es superior a 7 o inferior a 3.
   b) Minimizar la expresión algebraica de este circuito por el método de Karnaugh o numérico.
   c) Realizar la expresión mínima con puertas NO-Y y NO-O.
   d) Realizar este circuito con un multiplexor de ocho canales.
   e) Realizar este circuito con una memoria pasiva.
   f) Realizar este circuito con una matriz lógica programable (PLA) y una matriz lógica Y-programable (PAL).

2. Generar la función:

$$f = \prod_{4} (1, 3, 7, 8, 10) \prod_{\varnothing} (0, 5, 6, 14, 15)$$

con un decodificador y las puertas NO-O necesarias.

3. *a*) Obtener la expresión algebraica mínima de una función lógica de cuatro variables que toma el valor lógico uno cuando el número de variables que están en estado uno es superior al de las que se encuentran en estado cero. Nunca pueden estar más de tres variables en estado uno.
b) Realizar la expresión obtenida con puertas NO-O y NO-Y.
c) Obtener la expresión mínima de esta función con la función O-exclusiva.

4. *a*) Realizar la tabla de verdad de un convertidor del código BCD natural al BCD exceso tres.
b) Minimizar las expresiones algebraicas por el método tabular de Karnaugh y el método numérico.
c) Realizar este convertidor con puertas NO-Y y NO-O.
d) Realizar este convertidor con circuitos multiplexores.
e) Realizar este convertidor con una matriz lógica Y-programable (PAL).

5. Realizar un codificador con prioridad de 16 variables de entrada mediante el codificador de la figura 3.72 cuya tabla de verdad se representa en la tabla 3.23. Utilícense las puertas NO-Y que sean necesarias.

6. Realizar un decodificador del código ASCII de 6 bits (tabla 1.8) utilizando al máximo los circuitos decodificadores de escala de integración media.

7. En un registro de 6 bits cuyas salidas están disponibles en paralelo se almacena información en el código ASCII. Diseñar funciones lógicas que adopten el estado lógico uno, cuando la información contenida en dicho registro corresponda:

a) A un carácter numérico (0 al 9).
b) A un carácter alfabético.
c) A un carácter especial.

Realizar la síntesis de estas funciones en primer lugar con puertas NO-Y o NO-O y después con circuitos multiplexores.

8. a) Realizar por el método numérico la síntesis de la multifunción:

$$f_1 (a, b, c, d) = \prod_{4} (1, 2, 3, 4, 5, 12, 13, 14)$$

$$f_2 (a, b, c, d) = \prod_{4} (1, 2, 3, 7, 8, 9, 12, 13, 14)$$

$$f_3 (a, b, c, d) = \prod_{4} (0, 3, 8, 9)$$

b) Realizar el circuito con puertas NO-Y y NO-O.
c) Realizar el circuito utilizando el montaje «Y por conexión».
d) Realizar el circuito con una matriz lógica programable (PLA)

9. Diseñar un convertidor del código decimal (uno entre diez) al BCD natural con puertas NO-Y. Se supone que las entradas decimales son activas con un cero lógico.

Capítulo

4

Aritmética en los códigos binarios

4.1 GENERALIDADES

En la mayoría de los sistemas digitales, desde el circuito de control más sencillo hasta el computador más complejo, se realizan operaciones aritméticas. En este capítulo se estudian las operaciones aritméticas en los diferentes códigos binarios expuestos en el capítulo 1 y se diseñan circuitos aritméticos con circuitos integrados de escala de integración pequeña (SSI) y escala de integración media (MSI).

4.2 OPERACIONES EN EL SISTEMA BINARIO NATURAL

Las leyes que rigen las operaciones en el sistema de base dos son paralelas a las del sistema de base diez. En sucesivos apartados se analiza cada una de las operaciones elementales.

4.2.1 Suma aritmética binaria

La suma aritmética de dos bits resulta muy sencilla porque éstos sólo pueden tomar el valor cero y uno. La tabla de la suma en el sistema de base dos es:

Sumandos		*Suma binaria*	*Acarreo*
b	*a*	*S*	*C*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

La suma binaria toma el valor uno cuando uno solo de los sumandos tiene dicho valor. Cuando ambos sumandos tienen el valor uno, la suma es cero y se produce un acarreo.

De lo dicho se deduce que la suma aritmética de dos bits es una multifunción formada por dos funciones de dos variables de entrada y dos variables de salida, que son la suma binaria S y el acarreo C. La tabla de verdad de esta función se representa en la tabla 4.1.

b_0	a_0	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TABLA 4.1.—**Tabla de verdad del semisumador.**

De dicha tabla se deducen las expresiones algebraicas de S y C:

$$S = a_0\bar{b}_0 + \bar{a}_0 b_0 = \overline{a_0 b_0 + \bar{a}_0\bar{b}_0} = a_0 \oplus b_0$$

$$C = a_0 b_0 = \overline{\overline{a_0 b_0}} = \overline{\bar{a}_0 + \bar{b}_0}$$

En las ecuaciones algebraicas de S y C se observa que la suma binaria S es equivalente a la función O-exclusiva y el acarreo C al producto lógico. Al circuito lógico que realiza ambas funciones se le denomina semisumador porque sólo permite la suma de dos bits.

El circuito semisumador se puede realizar físicamente con diferentes funciones básicas. La forma más sencilla se representa en la figura 4.1 y está constituida por una puerta O-exclusiva que realiza la suma binaria y una puerta NO-O que realiza el acarreo (se supone que las variables a_0 y b_0 se tienen tanto en forma directa como inversa).

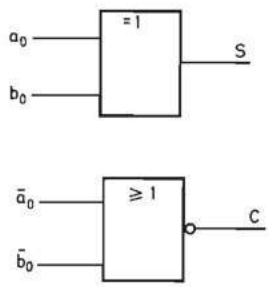


FIGURA 4.1.—**Circuito semisumador.**

En la figura 4.2 se representa la realización con puertas NO-O que se obtiene mediante transformación de las ecuaciones de S y C por los métodos estudiados en en el capítulo 3:

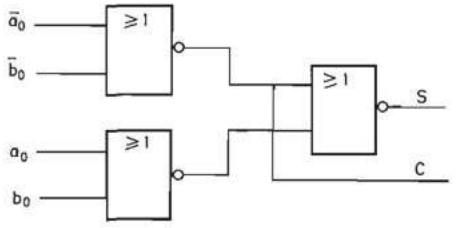


FIGURA 4.2.—**Circuito semisumador.**

$$S = \overline{a_0 b_0 + \bar{a}_0 \bar{b}_0} = (\bar{a}_0 + \bar{b}_0)(a_0 + b_0) = \overline{\overline{(\bar{a}_0 + \bar{b}_0)(a_0 + b_0)}} =$$

$$= \overline{\overline{\bar{a}_0 + \bar{b}_0} + \overline{a_0 + b_0}}$$

$$C = \overline{\bar{a}_0 + \bar{b}_0}$$

Cuando se desean sumar dos bits que forman parte de un número binario, es necesario sumar a ambos el acarreo procedente de la suma de los bits de peso inmediato inferior. El semisumador no permite realizar esta suma y para ello ha de diseñarse un nuevo sistema combinacional llamado sumador total, cuya tabla de verdad se indica en la tabla 4.2. C_n representa el acarreo procedente de la etapa

C_n	b	a	S	C_{n+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

TABLA 4.2.—**Tabla de verdad del sumador total.**

anterior y C_{n+1} el acarreo generado mediante la suma de los bits a y b y el acarreo C_n. De dicha tabla se deducen las expresiones de sumas canónicas de C_{n+1} y S:

$$S = a\bar{b}\bar{C}_n + \bar{a}b\bar{C}_n + \bar{a}\bar{b}C_n + abC_n$$
$$C_{n+1} = ab\bar{C}_n + a\bar{b}C_n + \bar{a}bC_n + abC_n$$

La expresión de S es equivalente a la función O-exclusiva de tres variables:

$$S = a \oplus b \oplus C_n$$

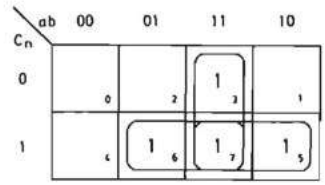


FIGURA 4.3

La expresión de C_{n+1} se puede simplificar al máximo mediante el método numérico o la tabla de Karnaugh indicada en la figura 4.3.

$$C_{n+1} = ab + aC_n + bC_n$$

El circuito sumador total puede realizarse con las puertas NO-Y, NO-O u O-exclusiva por los procedimientos indicados en el apartado 3.5 del capítulo 3. En la figura 4.4 se indica una realización posible de las expresiones algebraicas de S y C_{n+1} transformadas:

$$S = a \oplus b \oplus C_n$$

$$C_{n+1} = ab + aC_n + bC_n = ab + C_n(a + b) = \overline{\overline{ab + C_n(a + b)}} =$$

$$= \overline{\overline{ab}\ \ \overline{C_n(a + b)}} = \overline{\overline{ab}\ \ \overline{C_n\,\bar{a}\,\bar{b}}}$$

El sumador total es un circuito combinacional de aplicación general y por ello ha sido realizado en algunas tecnologías de circuitos integrados formando parte de la escala de integración media. Esto permite su utilización como bloque funcional y simplificar de esta forma el diseño de los sistemas digitales. En el apartado 4.2.2

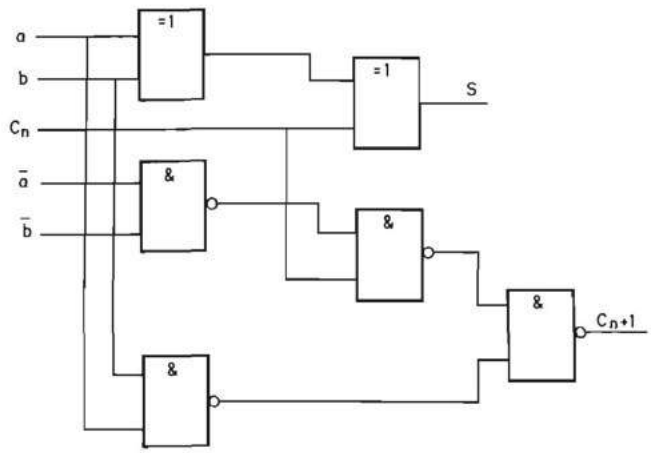


FIGURA 4.4.—**Circuito del sumador total.**

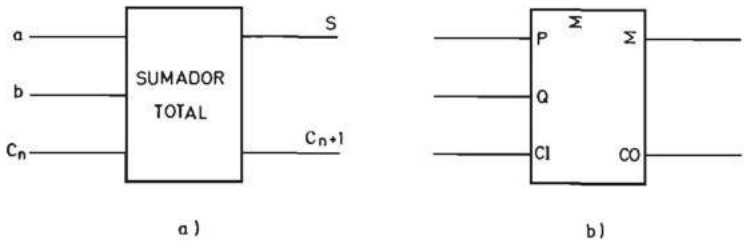


FIGURA 4.5.—**Esquema de bloques del sumador total.**

se estudian diversos circuitos sumadores de escala de integración media, sus características y aplicaciones.

En la figura 4.5*a* se indica el símbolo gráfico utilizado inicialmente para representar el circuito sumador y en la figura 4.5*b* el adoptado en el nuevo sistema de representación internacional.

En general ha de realizarse siempre la suma de números binarios de longitud superior a un bit. Dicha suma se puede realizar en serie o en paralelo.

El sumador serie realiza simultáneamente la suma de dos bits, uno de cada número y el acarreo procedente de la suma de los dos anteriores. Por tanto, está formado básicamente por un solo sumador total y ha de poseer un elemento que memorice el acarreo. A este circuito básico hay que añadirle registros para almacenar los operandos y el resultado, y una unidad de control que secuencia las sumas de dos bits. Se obtiene así un procesador digital secuencial síncrono a cuyo estudio se dedica el tomo II de este libro.

El sumador paralelo realiza simultáneamente la suma de dos números de n bits y para ello utiliza n sumadores totales. La suma en paralelo puede realizarse mediante dos procedimientos:

a) *Generación del acarreo en serie.* En este montaje cada sumador realiza la suma de dos bits y el acarreo procedente del sumador de los bits de peso inmediatamente inferior.

En la figura 4.6 se representa el esquema de bloques de un sumador de n bits de este tipo. En él se observa que el acarreo se propaga en serie de un sumador al siguiente y, por tanto, el tiempo necesario para que se realice la suma es igual a n veces el tiempo que tarda en generarse el acarreo en un sumador.

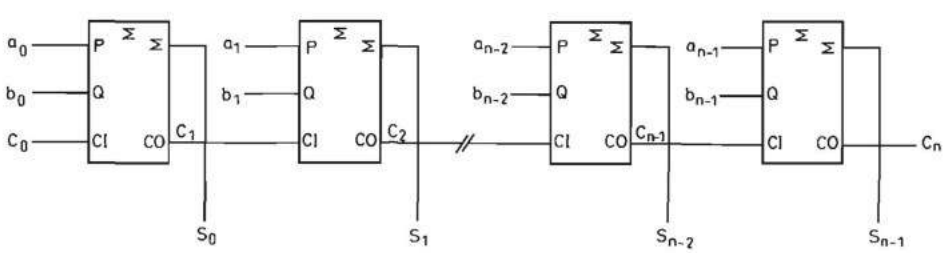


FIGURA 4.6.—**Esquema de bloques de un sumador paralelo de n bits con generación del acarreo en serie.**

Este montaje es adecuado para la realización de sumadores en los que el tiempo de operación no tiene que ser reducido al mínimo. Este sumador se puede realizar mediante circuitos de escala de integración media (MSI) tal como se indica en el apartado 4.2.2.

b) *Generación del acarreo en paralelo.* En este montaje los acarreos de todas las etapas son generados simultáneamente.

La expresión del acarreo de la etapa n en función del acarreo de la etapa anterior (n — 1) se dedujo en párrafos anteriores:

$$C_n = a_{n-1}b_{n-1} + (a_{n-1} + b_{n-1})\, C_{n-1}$$

De esta ecuación se deduce que C_n puede tomar el valor uno si los dos bits de la etapa a_{n-1} y b_{n-1} toman el valor uno simultáneamente o bien si cualquiera de ellos es uno y, a su vez, lo es también el acarreo de la etapa anterior. El término $a_{n-1}\, b_{n-1}$ genera un acarreo en la propia etapa y se denomina generador G_n *[Carry Generator (CG)]*. El término $(a_{n-1} + b_{n-1})$ permite la propagación del acarreo de la etapa anterior y se llama propagador P [*Carry Propagator* (*CP*)].

Por lo tanto, C_n se puede indicar:

$$C_n = G_{n-1} + P_{n-1}\, C_{n-1}$$

Seguidamente se deduce la ecuación del acarreo en paralelo de n etapas de un sumador de n bits. Si se denomina C_0 al acarreo que se aplica a la primera etapa, resulta:

$$C_1 = G_0 + P_0 C_e$$

En la segunda etapa:

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_e$$

En la tercera etapa:

$$\begin{aligned} C_3 = G_2 + P_2 C_2 &= G_2 + P_2\,(G_1 + P_1\, G_0 + P_1 P_0\, C_e) = \\ &= G_2 + P_2\, G_1 + P_2\, P_1 G_0 + P_2 P_1 P_0 C_e \end{aligned}$$

En la cuarta etapa:

$$\begin{aligned} C_4 = G_3 + P_3 C_3 &= G_3 + P_3\,(G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_e) = \\ &= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_e \end{aligned}$$

En la etapa enésima se tiene, por tanto:

$$C_n = G_{n-1} + P_{n-1}G_{n-2} + P_{n-1}P_{n-2}G_{n-3} + \ldots + P_{n-1}P_{n-2} \ldots P_1\, G_0 + P_{n-1}P_{n-2} \ldots P_1 P_0 C_e$$

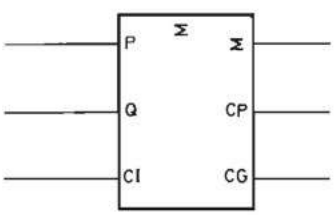


FIGURA 4.7.—**Símbolo lógico del sumador total con salidas de propagación y generación del acarreo.**

La utilización de los términos de generación del acarreo *G (CG)* y propagación del acarreo *P (CP)*, hace interesante representar el sumador mediante el símbolo indicado en la figura 4.7, y utilizarlo como bloque funcional para obtener el diagrama de bloques de un sumador de cuatro bits con generación del acarreo en paralelo que se representa en la figura 4.8.

En escala de integración media se han realizado los cuatro bloques *CPG* de la figura 4.8 en un solo circuito integrado, constituyendo un bloque funcional denominado generador de acarreos que reduce la complejidad de los sumadores. En la

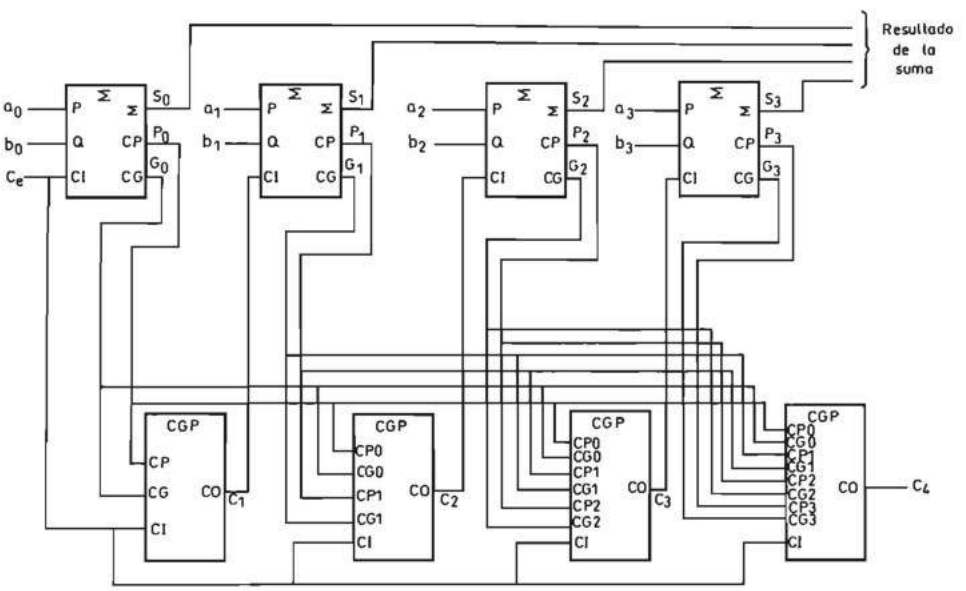


FIGURA 4.8.—**Sumador en paralelo con generación del acarreo en paralelo.**

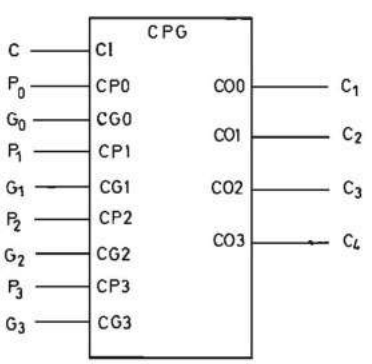


FIGURA 4.9.—**Diagrama de bloques de un generador de acarreos.**

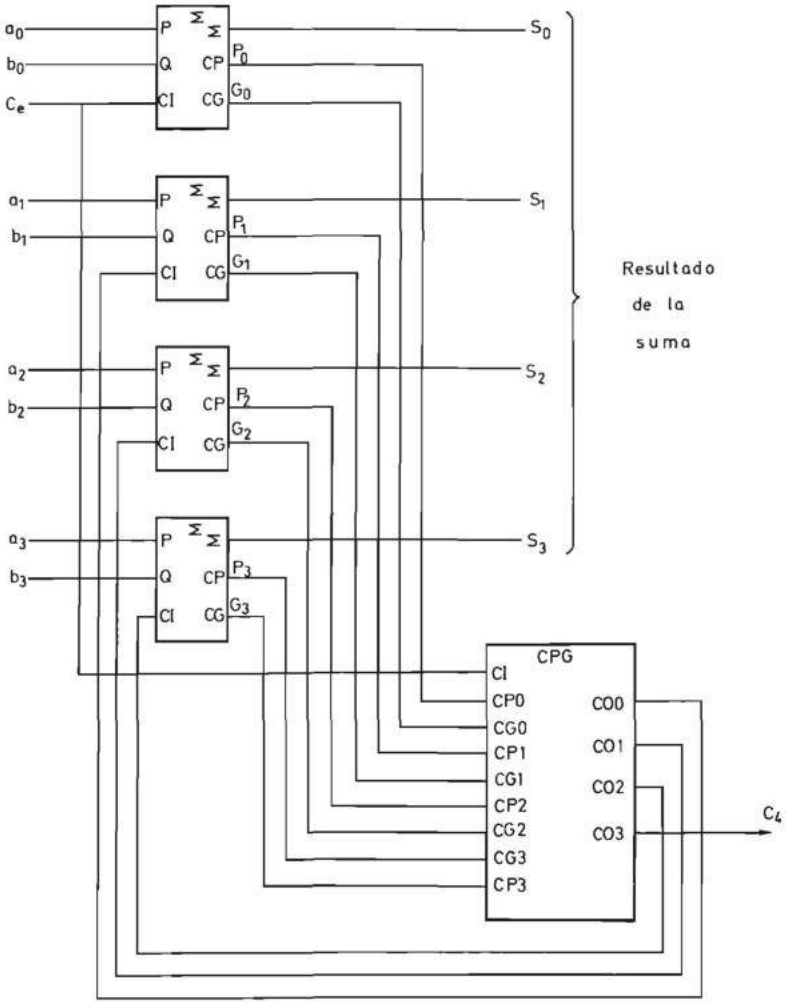


FIGURA 4.10.—**Sumador en paralelo con generación del acarreo en paralelo.**

figura 4.9 se representa el diagrama de bloques de este circuito. En la figura 4.10 se representa el esquema del sumador con generación del acarreo en paralelo realizado con este bloque funcional.

A su vez el sistema de la figura 4.10 constituye un bloque funcional, que se representa en la figura 4.11 [el indicativo *(CPG)* debajo del símbolo Σ, indica que se trata de un sumador con generación del acarreo en paralelo].

Para obtener un sumador de mayor capacidad es posible conectar entre sí bloques como el de la figura 4.11 en un montaje de propagación en serie tal como se indica en la figura 4.12, que representa un sumador de 16 bits.

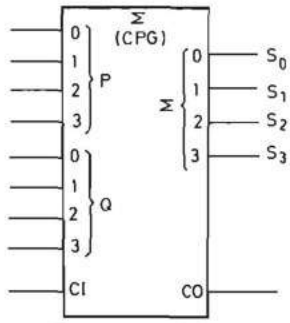


FIGURA 4.11.—**Símbolo lógico de un sumador de cuatro bits en paralelo con generación del acarreo en paralelo.**

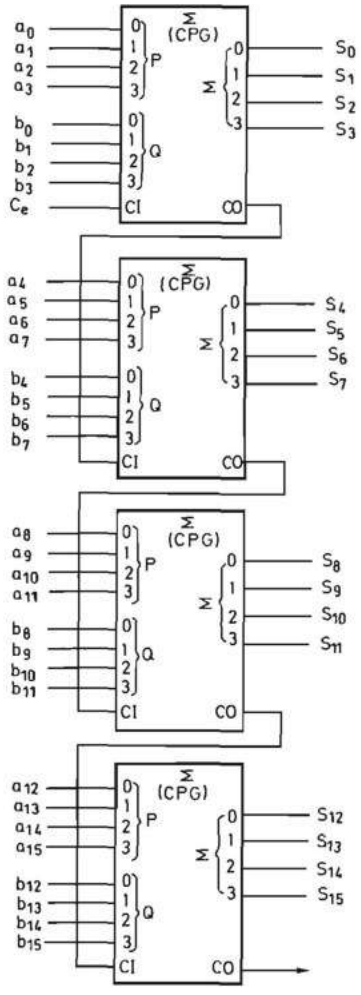


FIGURA 4.12.—**Sumador de dieciséis bits realizado con cuatro bloques idénticos al de la figura 4.11, conectados con propagación del acarreo en serie.**

Pero es posible obtener un sumador más rápido si se utiliza un generador de acarreo en paralelo.

En efecto, el acarreo C_4 del circuito de la figura 4.9 tiene, de acuerdo con la expresión de C_n obtenida anteriormente, la ecuación:

$$C_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_e$$

En esta expresión se puede llamar *G (CG)* a la expresión $G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$. Igualmente se puede llamar *P (CP)* a $P_3P_2P_1P_0$. Se obtiene así:

$$C_4 = G + P\,C_e$$

Es posible por ello realizar un generador de acarreo que, en lugar de C_4, dé a su salida *G* y *P*, tal como se representa en la figura 4.13. El lector puede deducir el esquema de un sumador de cuatro bits que utilice este circuito sin más que sustituir el *CPG* de la figura 4.10 por el que se acaba de desarrollar. En la figura 4.14 se representa el símbolo lógico de un sumador de cuatro bits realizado con el *CPG* de la figura 4.13. A partir de él se puede obtener el sumador de 16 bits de la figura 4.15 que combina el sumador de la figura 4.14 con el generador de acarreos de

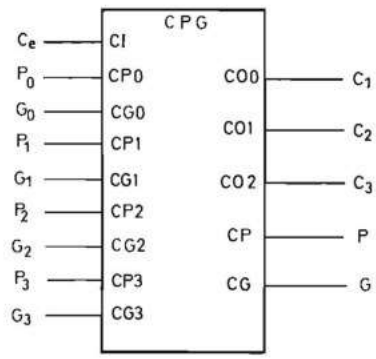


FIGURA 4.13.—**Símbolo lógico de un circuito generador de acarreo de cuatro bits con salidas de propagación y generación de acarreos.**

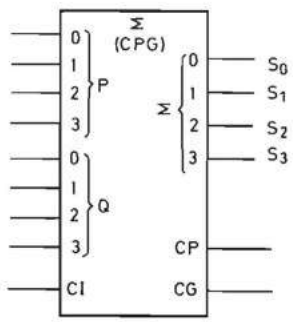


FIGURA 4.14.—**Símbolo lógico de un sumador de cuatro bits con generación del acarreo en paralelo realizado con el circuito de la figura 4.13.**

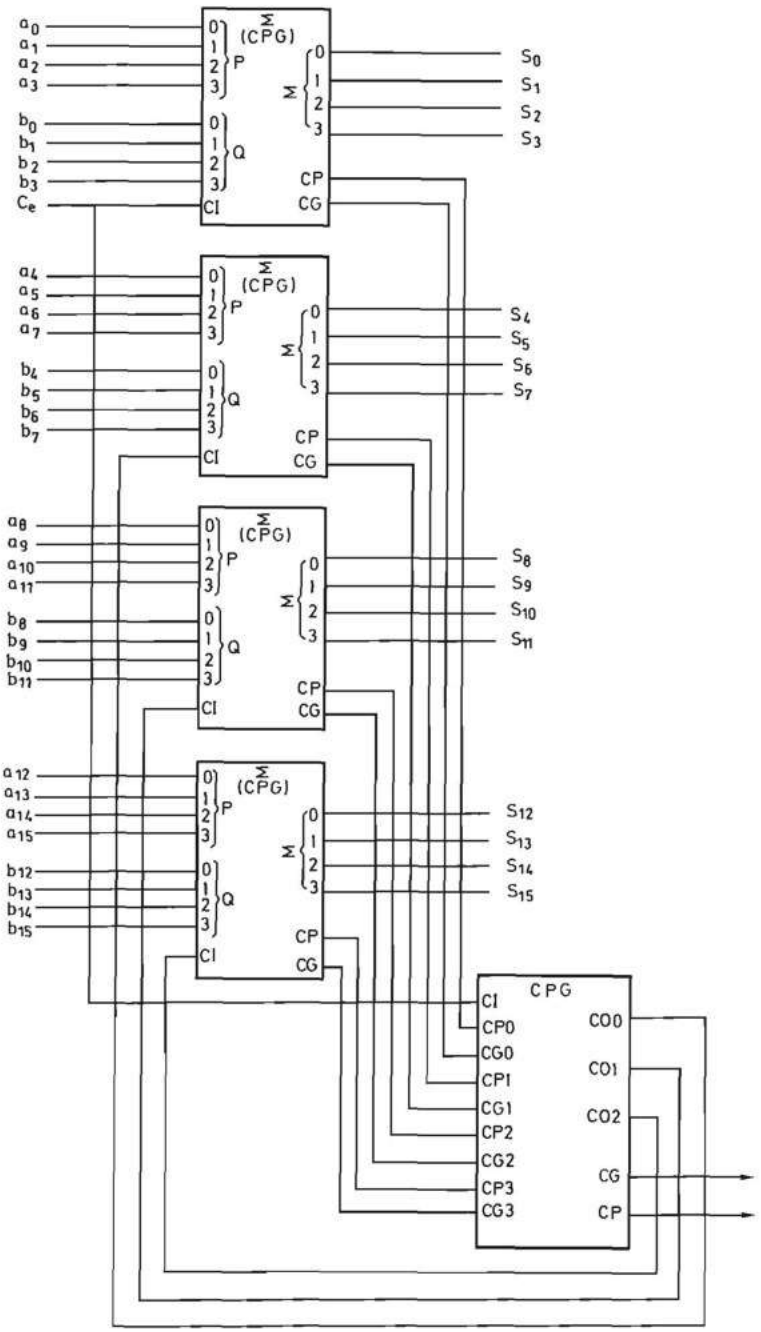


FIGURA 4.15.—**Sumador de dieciséis bits realizado con el sumador de la figura 4.14 y el circuito generador de acarreo de la figura 4.13.**

la figura 4.13. Comparándolo con el de la figura 4.12, el lector puede deducir que su velocidad es indudablemente mayor.

A su vez existen soluciones que representan un término medio entre la generación del acarreo en serie y en paralelo, que aunque hayan perdido interés práctico debido al progreso de la microelectrónica, es interesante que el lector analice para comprender las numerosas alternativas que surgen en la realización de los circuitos digitales cuando se eleva su complejidad.

En la figura 4.16 se representa un sumador de 12 bits realizado mediante tres sumadores con propagación del acarreo en serie de cuatro bits (se indica dicha circunstancia colocando el apelativo serie debajo del símbolo Σ).

La propagación del acarreo se realiza en serie en el interior del cuádruple sumador, y cada cuatro bits se genera o propaga el acarreo exteriormente. El circuito generador de acarreo recibe directamente los números a sumar y genera el acarreo y su inverso.

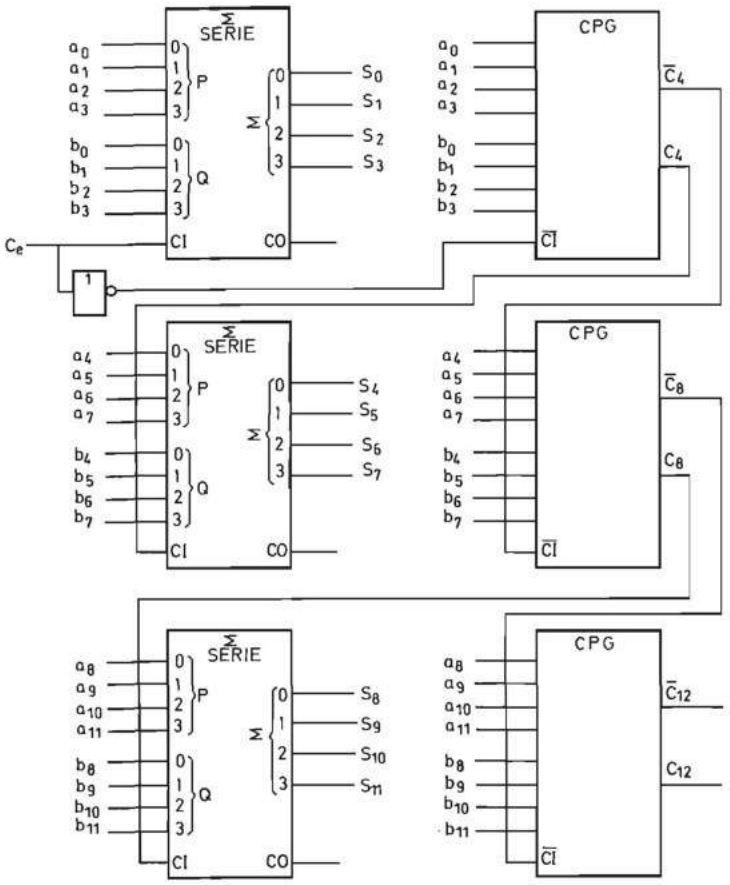


FIGURA 4.16.—**Sumador de 12 bits realizado con tres sumadores de cuatro bits con propagación del acarreo en serie, conectados con propagación del acarreo en paralelo.**

A continuación se realiza su diseño. La expresión algebraica de C_4 es:

$$C_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_e$$

Invirtiendo los dos miembros de la igualdad resulta:

$$\overline{C_4} = \overline{G_3}\ \overline{P_3G_2}\ \overline{P_3P_2G_1}\ \overline{P_3P_2P_1G_0}\ \overline{P_3P_2P_1P_0C_e}$$

$$\overline{C_4} = \overline{G_3}\ (\overline{P_3} + \overline{G_2})\ (\overline{P_3} + \overline{P_2} + \overline{G_1})\ (\overline{P_3} + \overline{P_2} + \overline{P_1} + \overline{G_0})\ (\overline{P_3} + \overline{P_2} + \overline{P_1} + \overline{P_0} + \overline{C_e})$$

$$\overline{C_4} = \overline{G_3}\overline{P_3} + \overline{G_3}\overline{G_2}\overline{P_2} + \overline{G_3}\overline{G_2}\overline{G_1}\overline{P_1} + \overline{G_3}\overline{G_2}\overline{G_1}\overline{G_0}\overline{P_0} + \overline{G_3}\overline{G_2}\overline{G_1}\overline{G_0}\overline{C_e}$$

Recordando que:

$$G_i = a_ib_i$$
$$P_i = a_i + b_i$$

y que:

$$\bar{G}_i\bar{P}_i = \overline{a_ib_i}\ \ \overline{a_i + b_i} = (\bar{a}_i + \bar{b}_i)\,\bar{a}_i\bar{b}_i = \bar{a}_i\bar{b}_i = \overline{a_i + b_i}$$

Sustituyendo estas expresiones en la ecuación de $\bar{C}_4$ resulta:

$$C_4 = \overline{a_3 + b_3} + \overline{a_3b_3}\ \overline{a_2 + b_2} + \overline{a_3b_3}\ \overline{a_2b_2}\ \overline{a_1 + b_1} +$$
$$+ \overline{a_3b_3}\ \overline{a_2b_2}\ \overline{a_1b_1}\ \overline{a_0 + b_0} + \overline{a_3b_3}\ \overline{a_2b_2}\ \overline{a_1b_1}\ \overline{a_0b_0C_e}$$

Aplicando una doble inversión al segundo miembro de esta expresión resulta:

$$\overline{C_4} = \overline{\overline{a_3 + b_3} + \overline{a_3b_3}\ \overline{a_2 + b_2} + \overline{a_3b_3}\ \overline{a_2b_2}\ \overline{a_1 + b_1} +}$$
$$\overline{+ \overline{a_3b_3}\ \overline{a_2b_2}\ \overline{a_1b_1}\ \overline{a_0 + b_0} + \overline{a_3b_3}\ \overline{a_2b_2}\ \overline{a_1b_1}\ \overline{a_0b_0C_e}}$$

$$\overline{C_4} = \overline{\overline{a_3b_3}\ \overline{a_3b_3}\ \overline{a_2 + b_2}\ \overline{a_3b_3}\ \overline{a_2b_2}\ \overline{a_1 + b_1} \cdot}$$
$$\overline{\cdot\ \overline{a_3b_3}\ \overline{a_2b_2}\ \overline{a_1b_1}\ \overline{a_0 + b_0}\ \overline{a_3b_3}\ \overline{a_2b_2}\ \overline{a_1b_1}\ \overline{a_0b_0C_e}}$$

En la figura 4.17 se representa el esquema de este circuito.

4.2.2 El circuito sumador total como bloque funcional

El desarrollo de la microelectrónica propició la integración de los sumadores totales en circuitos de escala de integración media (MSI), lo cual permitió su utilización como bloque funcional y simplificó, de esta forma, el diseño de los sistemas digitales.

El desarrollo de los circuitos de gran escala de integración (LSI) hizo que los sumadores se integrasen junto con conjuntos de registros (ver apartado 6.3.5.6.2) para constituir la unidad operativa de un procesador digital secuencial.

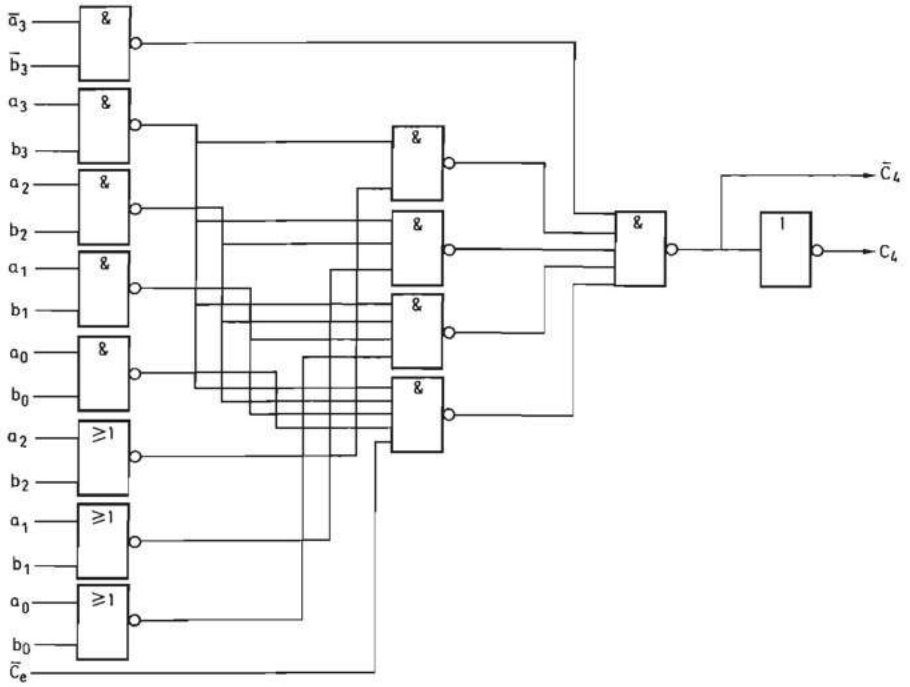


FIGURA 4.17.—**Esquema del circuito generador de acarreo de la figura 4.16.**

La fabricación de los circuitos integrados semimedida (Semicustom) y totalmente a medida, que dan al diseñador de sistemas digitales la posibilidad de diseñar sus propios circuitos integrados, hace interesante el estudio de los circuitos sumadores y sus diferentes formas de realización.

En escala de integración media (MSI) existen circuitos que contienen uno o más sumadores totales y en el segundo caso se pueden clasificar en dos categorías:

a) Los sumadores totales contenidos en el circuito integrado son totalmente independientes y todas sus entradas y salidas son accesibles al exterior.

b) Los sumadores totales contenidos en el circuito integrado están conectados entre sí de forma que la salida de acarreo de uno se conecta a la entrada correspondiente del otro. Solamente son accesibles al exterior la entrada de acarreo del primer sumador y la salida de acarreo del último, además de los bits que se desean sumar en cada sumador. Estos circuitos constituyen un sumador de tantos bits como sumadores totales contienen, con propagación del acarreo en serie.

La utilización de sumadores totales independientes permite la realización de la suma de números binarios de n bits con acarreo anticipado, tal como se ha explicado en el apartado anterior.

En la figura 4.18 se representa un sumador de 2 bits con propagación del acarreo en serie junto a sus símbolos lógicos no normalizado y normalizado.

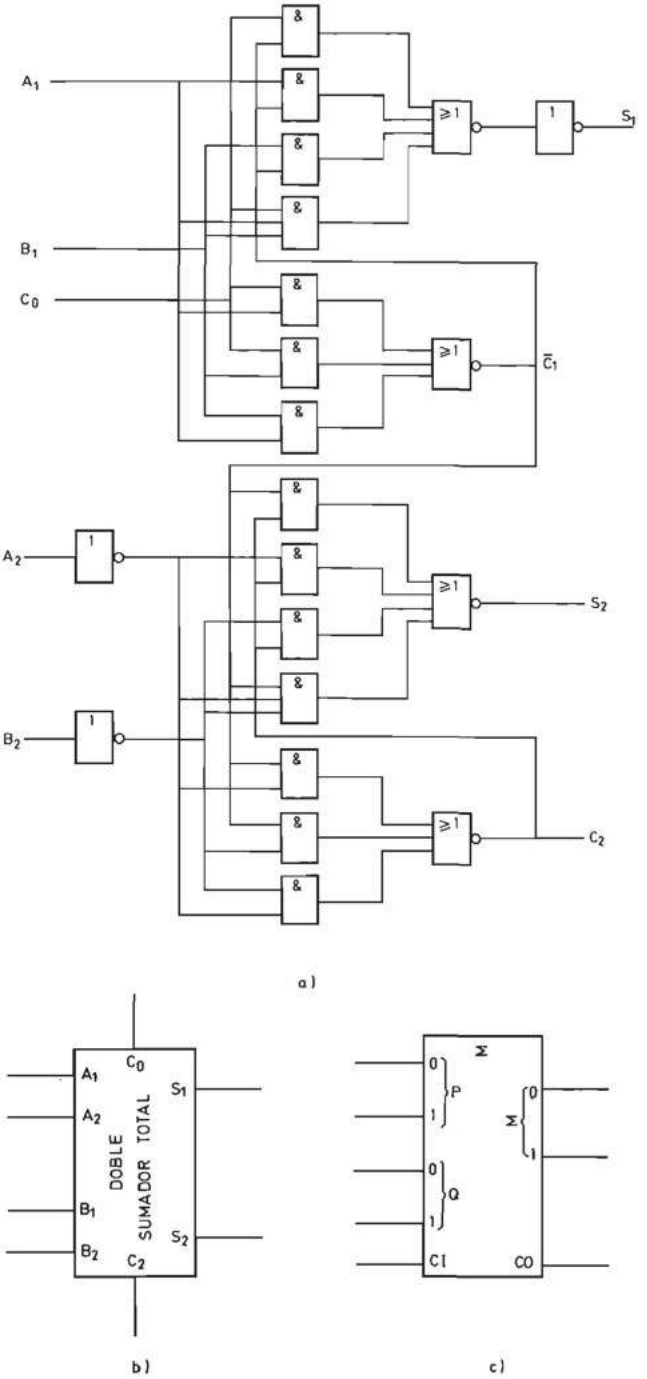


FIGURA 4.18.—**Sumador de dos bits con propagación del acarreo en serie: *a*) esquema del circuito; *b*) símbolo lógico no normalizado; *c*) símbolo lógico normalizado.**

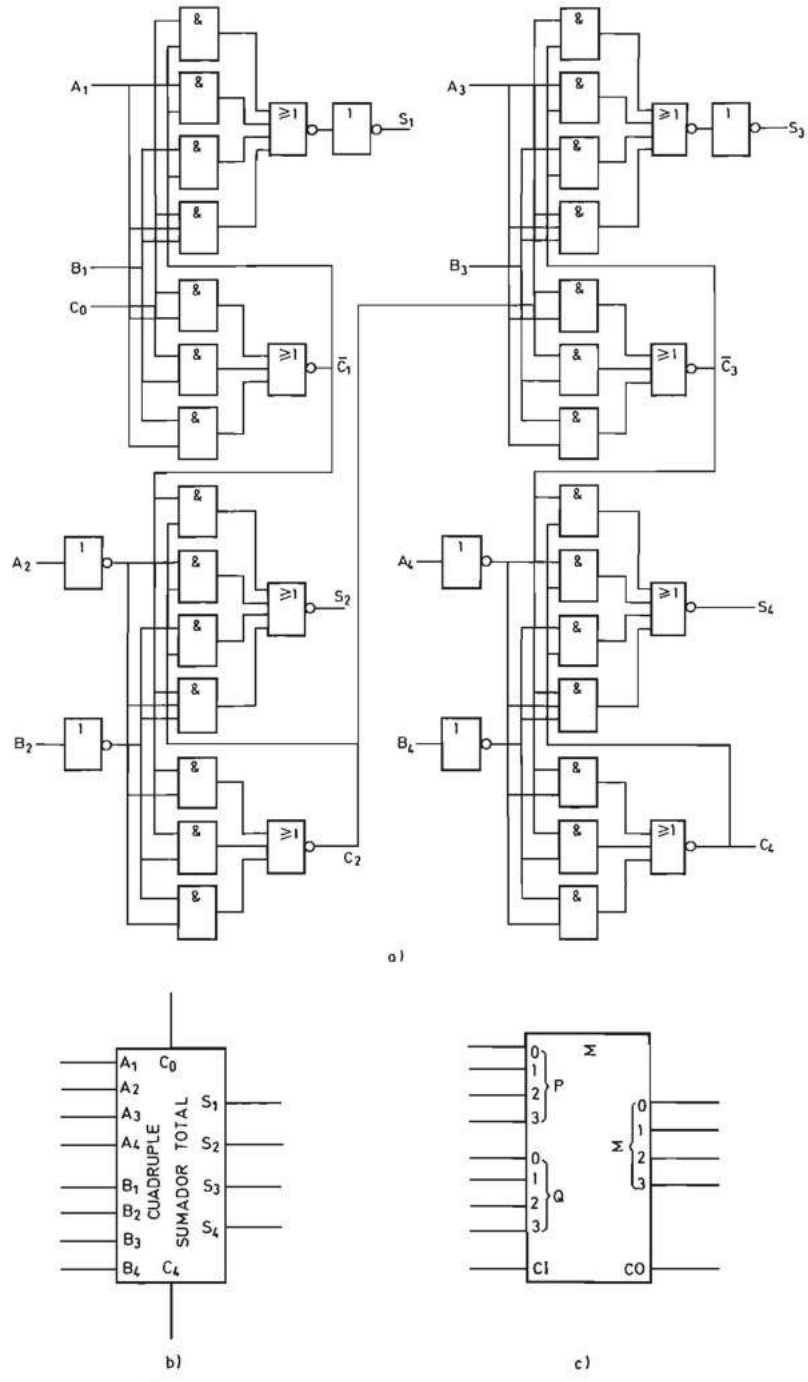


FIGURA 4.19.—**Sumador total de cuatro bits con propagación del acarreo en serie: *a*) esquema del circuito; *b*) símbolo lógico no normalizado; *c*) símbolo lógico normalizado.**

En la figura 4.19 se representa un sumador de números binarios de 4 bits con propagación del acarreo en serie, junto a sus símbolos lógicos no normalizado y normalizado.

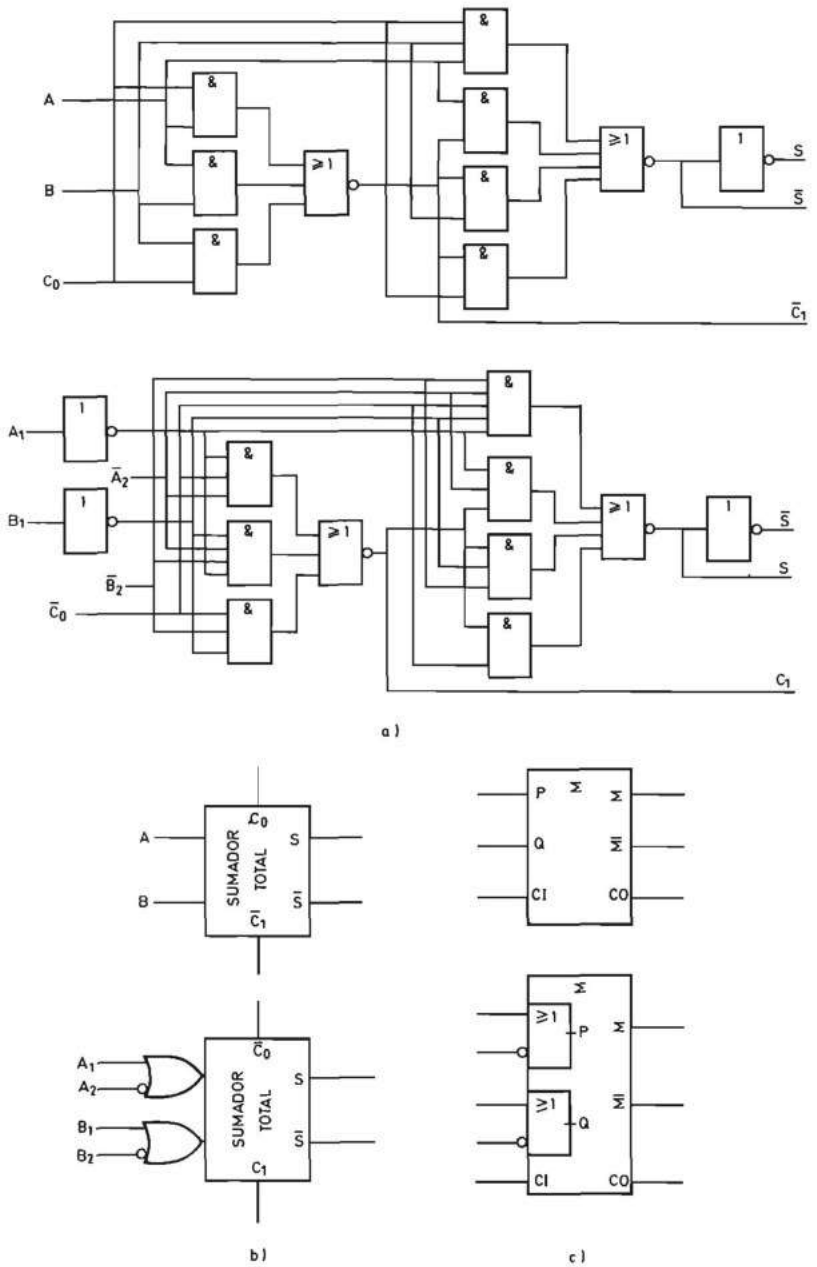


FIGURA 4.20.—**Doble sumador de un bit: *a*) esquema del circuito; *b*) símbolos lógicos no normalizados; *c*) símbolos lógicos normalizados.**

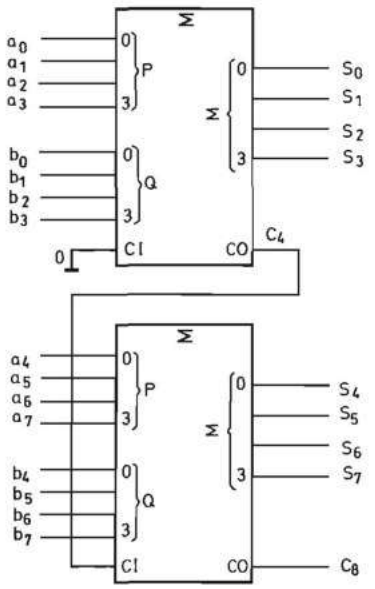


FIGURA 4.21.—**Sumador de ocho bits realizado con dos sumadores de cuatro bits con propagación del acarreo en serie.**

En la figura 4.20 se representa un circuito que contiene dos sumadores totales independientes.

En la figura 4.21 se representa el esquema de un sumador de números binarios de 8 bits realizado mediante dos cuádruples sumadores como el de la figura 4.19.

Estos circuitos sumadores tienen numerosas aplicaciones además de la realización de operaciones aritméticas como son, entre otras, la conversión de códigos, comparación binaria y generación de paridad, de las cuales se exponen seguidamente algunos ejemplos.

En la figura 4.22 se indica la forma de realizar un convertidor de BCD natural a BCD exceso tres, utilizando el cuádruple sumador total. El número a convertir se aplica a las entradas A del sumador y las entradas B se aplica el número 0011. Las salidas del sumador son el equivalente en BCD exceso tres del número BCD natural aplicado a las entradas A_1 a A_4. Esta conversión se realiza en algunos microprocesadores y procesadores digitales de señales para realizar operaciones con números codificados en BCD natural mediante un sumador binario natural.

En la figura 4.23 se representa el esquema de un comparador binario de números de cuatro bits realizado con el cuádruple sumador total. La comparación se realiza sumando a un número el inverso del otro. La salida del sumador es 1111 solamente cuando ambos números son iguales.

Cuando la combinación de salida del sumador es distinta del 1111, el valor lógico del acarreo C_4 indica cuál de los dos números de entrada es mayor. Si el número que no se invierte es mayor que el otro, el acarreo es igual a uno y, en caso contrario, es igual a cero.

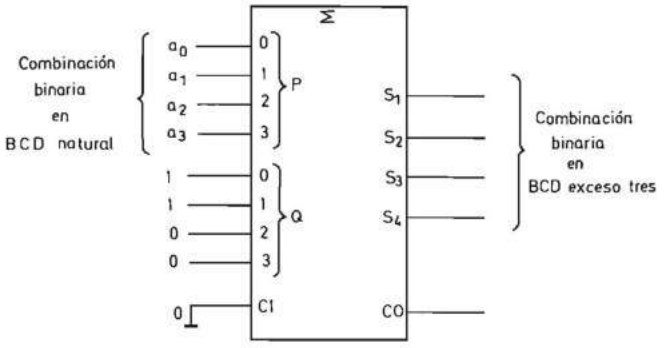


FIGURA 4.22.—**Convertidor de BCD natural a BCD exceso tres realizado con un sumador de cuatro bits.**

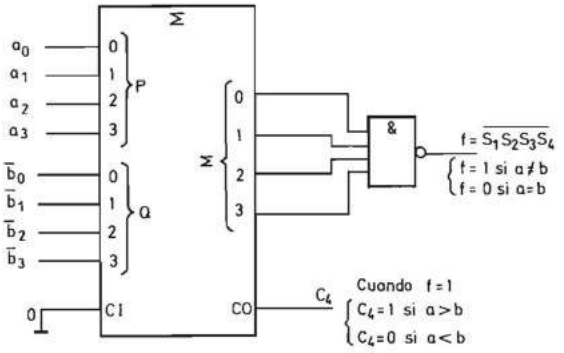


FIGURA 4.23.—**Comparador de cuatro bits realizado con un sumador de cuatro bits y una puerta NO-Y (NAND).**

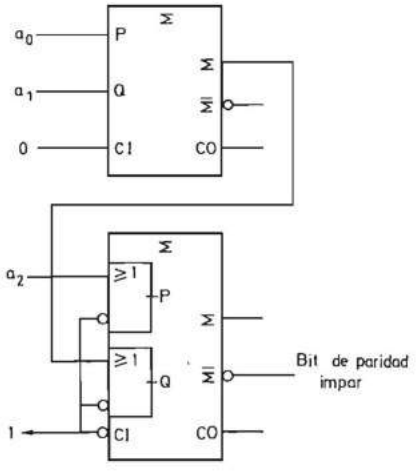


FIGURA 4.24.—**Circuito generador de paridad impar de un número de tres bits realizado con el doble sumador de un bit de la figura 4.20.**

En la figura 4.24 se representa un generador de paridad para un número de tres bits realizado con el doble sumador de 1 bit de la figura 4.20. Para generar el bit de paridad se suman todos los bits del número binario (sin tener en cuenta el acarreo) y el resultado se agrega al mismo como un bit más. El circuito diseñado en la figura 4.24 genera un bit de paridad impar, es decir, cuando el número binario tenga un número par de unos, el bit de paridad generado es un uno.

4.2.3 Resta binaria

La resta de dos números binarios es equivalente a una suma algebraica. Por ejemplo, restar el número positivo b del número positivo a es equivalente a sumar al número a el número $-b$. Por tanto, si los números negativos se representan adecuadamente, la resta puede ser convertida en una suma y realizada mediante un circuito sumador.

Antes de estudiar la resta binaria se analiza a continuación la representación de los números negativos.

4.2.3.1 Representación de los números negativos. Un convenio que ha sido utilizado en todos los sistemas de numeración para la representación de los números negativos es el de emplear una cifra en el extremo izquierdo del número correspondiente para representar el signo. En el sistema binario objeto de nuestro estudio dicha cifra, llamada bit de signo, es 0 si el número es positivo y 1 si es negativo. En este sistema denominado de valor absoluto y signo porque los números negativos se representan mediante su valor absoluto y el signo, se utilizan n bits para indicar los números binarios de n — 1 bits. Este método de representación de los números negativos implica que, para la realización de la resta es necesario utilizar un circuito restador distinto del circuito sumador estudiado en el apartado 4.2 y, por tanto, una mayor complejidad del circuito lógico capaz de realizar ambas operaciones. Por esto, el convenio de disponer de un bit de signo se combina con el de representar un número negativo de n bits (n — 1 bits de información más un bit de signo) mediante el complemento a la base dos y complemento a la base menos uno (uno). Mediante la adopción de estos dos convenios, tal como veremos en sucesivos apartados, la operación de la resta se convierte en suma y, por tanto, mediante un único circuito sumador se realizan ambas operaciones, lo cual simplifica enormemente la realización física de la unidad aritmética.

El complemento a dos de un número a de n bits se obtiene restando el número de 2^n (en binario, 1 seguido de n ceros), lo cual es equivalente a complementarlo, es decir, cambiar, los unos por ceros y viceversa, y sumándole la unidad.

Como ejemplo se representa el número +19 en el sistema binario mediante cinco bits ($2^4 < 19 < 2^5$) más un bit de signo:

$$19_{10} \equiv \underset{\substack{\textit{Bit de}\\ \textit{signo}}}{0}\ 1\ 0\ 0\ 1\ 1$$

El bit de signo se indica mediante las siglas *B.S.*

El número -19 se representará mediante el complemento a dos:

$$\begin{array}{rr} & 1\ 0\ 0\ 0\ 0\ 0\ 0 \\ - & \quad 0\ 1\ 0\ 0\ 1\ 1 \\ \hline -19 \equiv & \quad 1\ 0\ 1\ 1\ 0\ 1 \end{array}$$

El lector puede comprobar que se obtiene el mismo resultado complementando 19 y sumándole la unidad.

Otro ejemplo:

$$\begin{array}{rc} & \textit{Bit de signo} \\ +27_{10} \equiv & \quad 0\ \ 1\ 1\ 0\ 1\ 1 \\ & \textit{Bit de signo} \\ -27_{10} \equiv & \quad 1\ \ 0\ 0\ 1\ 0\ 1 \end{array}$$

El complemento a uno de un número de n bits se obtiene restando el número de $2^n - 1$ (en binario $\overbrace{1 \ldots 1}^{n}$) lo cual es equivalente a complementarlo, es decir, cambiar los unos del número por ceros y viceversa.

A continuación se representan en el convenio del complemento a uno los números que antes se han indicado mediante el complemento a dos. El lector puede comprobar los resultados:

$$\begin{array}{rc} & \textit{Bit de signo} \\ -19_{10} \equiv & \quad 1\ \ 0\ 1\ 1\ 0\ 0 \\ -27_{10} \equiv & \quad 1\ \ 0\ 0\ 1\ 0\ 0 \end{array}$$

4.2.3.2 Operación con números negativos mediante el convenio del complemento a dos. Estudiaremos la resta del número b del número a, ambos de n bits (n — 1 bits de información y un bit de signo). Los casos que se pueden presentar son los siguientes:

1) a y b son ambos positivos.

Representando $-b$ por su complemento a 2^n, para restar b de a, realizaremos la operación:

$$a - b \equiv a + (2^n - b) = 2^n + (a - b)$$

Si $a > b$ se producirá un acarreo que no es tenido en cuenta por ser el resultado positivo y quedar representado en forma directa.

$$\begin{array}{rc} & \textit{B.S.} \\ a = 25_{10} \equiv & \quad 0\ \ 1\ 1\ 0\ 0\ 1_2 \\ b = 18_{10} \equiv & \quad 0\ \ 1\ 0\ 0\ 1\ 0_2 \end{array}$$

$-b$ se representa mediante el complemento a dos:

$$\begin{array}{rc} & \textit{B.S.} \\ -b = -18_{10} \equiv & \quad 1\ \ 0\ 1\ 1\ 1\ 0 \end{array}$$

La diferencia 25 − 18 resulta:

		B.S.
25		0 1 1 0 0 1
−18	+	1 0 1 1 1 0
7		1 0 0 0 1 1 1

Despreciando el acarreo producido tenemos como resultado:

B.S.
$0\ 0\ 0\ 1\ 1\ 1 \equiv 7_{10}$

Si $a < b$, no se produce acarreo y, debido a que $a - b$ es negativo, tendremos:

$$a + (2^n - b) = 2^n - (b - a)$$

es decir, el resultado vendrá representado mediante su complemento a 2^n.

Ejemplo:

B.S.
$a = 18_{10} \equiv 0\ 1\ 0\ 0\ 1\ 0_2$
$b = 25_{10} \equiv 0\ 1\ 1\ 0\ 0\ 1_2$

$-b$ se representa:

B.S.
$1\ 0\ 0\ 1\ 1\ 1_2$

La diferencia 18 − 25 resulta:

		B.S.
18		0 1 0 0 1 0
−25	+	1 0 0 1 1 1
− 7		1 1 1 0 0 1

El lector puede comprobar que este resultado es el complemento a 2^n de

$$7_{10} \equiv 000111_2$$

2) a y b son ambos negativos.

La diferencia entre $-a$ y $-b$ resulta

$$(-a) - (-b) \equiv (2^n - a) + [2^n - (2^n - b)] =$$
$$= 2^n - a + 2^n - 2^n + b = 2^n + (b - a) =$$
$$= 2^n - (a - b)$$

Al igual que en el caso anterior, si el resultado es positivo se produce un acarreo que no es tenido en cuenta y, si el resultado es negativo, queda representado mediante su complemento a dos.

Ejemplos:

B.S.
$(-a) = -20_{10} \equiv 1\ 0\ 1\ 1\ 0\ 0_2$
$(-b) = -24_{10} \equiv 1\ 0\ 0\ 0\ 0\ 0_2$
$-(-b) = 24_{10} \equiv 0\ 1\ 1\ 0\ 0\ 0_2$

		B.S.
	−20	1 0 1 1 0 0
−	−24	+ 0 1 1 0 0 0
	4	1 0 0 0 1 0 0

Despreciando el acarreo resulta

$$4_{10} \equiv \underset{B.S.}{0\ 0\ 0\ 1\ 0\ 0_2}$$

Si se tiene

		B.S.
$(-a) = -24_{10} \equiv$		1 0 1 0 0 0_2
$(-b) = -20_{10} \equiv$		1 0 1 1 0 0_2
$-(-b) = 20_{10} \equiv$		0 1 0 1 0 0_2

		B.S.
	−24	1 0 1 0 0 0
−	−20	+0 1 0 1 0 0
	−4	1 1 1 1 0 0

El lector puede comprobar que este resultado es el complemento a 2^n de $4_{10} \equiv 000100$.

3) a es positivo y b negativo.

La diferencia $a - (-b)$ da como resultado un número positivo y no se produce acarreo.

En efecto:

$$a - (-b) \equiv a + 2^n - (2^n - b) = a + b$$

Ejemplo:

		B.S.
$a =$	$14_{10} \equiv$	0 0 1 1 1 0_2
$b =$	$17_{10} \equiv$	0 1 0 0 0 1_2
$-b = -17_{10} \equiv$		1 0 1 1 1 1_2

		B.S.
	14	0 0 1 1 1 0
−	−17	+0 1 0 0 0 1
	31	0 1 1 1 1 1

4) a es negativo y b positivo.

La diferencia $-a - b$ da como resultado un número negativo que queda representado mediante su complemento a 2^n. Además se produce un acarreo que no es tenido en cuenta.

En efecto:

$$-a - b \equiv (2^n - a) + (2^n - b) = 2^n + [2^n - (a + b)]$$

Ejemplo:

$$\begin{array}{rrl} & & B.S. \\ a = & 12_{10} \equiv & 0\ 0\ 1\ 1\ 0\ 0_2 \\ b = & 17_{10} \equiv & 0\ 1\ 0\ 0\ 0\ 1_2 \\ -a = & -12_{10} \equiv & 1\ 1\ 0\ 1\ 0\ 0_2 \\ -b = & -17_{10} \equiv & 1\ 0\ 1\ 1\ 1\ 1_2 \end{array}$$

$$\begin{array}{rrl} & & B.S. \\ -12 & & 1\ 1\ 0\ 1\ 0\ 0 \\ -17 & & +1\ 0\ 1\ 1\ 1\ 1 \\ \hline -29 & & 1\ 1\ 0\ 0\ 0\ 1\ 1 \end{array}$$

Despreciando el acarreo resulta:

$$\begin{array}{rrl} & & B.S. \\ -29_{10} \equiv & & 1\ 0\ 0\ 0\ 1\ 1 \end{array}$$

Por tanto queda demostrado que, mediante el convenio del complemento a 2^n, la resta se reduce a una suma.

Seguidamente se diseña un sumador-restador de 7 bits y signo. Se supone que los números negativos se presentan a su entrada representados en el convenio del complemento a 2^n.

En la figura 4.25 se representa el circuito del sumador-restador cuya parte aritmética está constituida por ocho sumadores totales con propagación en serie del acarreo. Para su realización se utilizan dos circuitos integrados como el indicado en la figura 4.19. Si la operación a realizar es la suma, se han de presentar en las entradas de todos los sumadores ambos números a y b. Si se desea realizar la resta de b del número a, se ha de obtener el complemento a 2^n de aquél para lo cual se ha de complementar y sumar uno al bit menos significativo. Para ello se aplica el número b a las entradas del sumador a través de puertas O-exclusiva de dos entradas controladas por la señal $\bar{S}/R$. Si el estado de $\bar{S}/R$ es un cero lógico, a la salida de las puertas O-exclusiva aparece el número b sin complementar y el bit de acarreo de entrada del primer sumador es un cero lógico, con lo cual se realiza la operación $a + b$. Si, por el contrario, el estado de $\bar{S}/R$ es un uno lógico, a las salidas de las puertas O-exclusiva aparece el inverso del número b y el bit de acarreo de entrada del primer sumador es un uno lógico, lo cual implica que se realice la operación $a - b$.

Si el valor absoluto del resultado es igual o superior a $2^7 = 128_{10}$, se produce un rebasamiento de la máxima capacidad del sumador. Este rebasamiento es sólo posible si los números que se presentan a las entradas de ambos sumadores en la operación de la resta o de la suma son ambos positivos o negativos. Para detectarlo se utilizan los bits de signo de ambos números y el acarreo que se suma a ambos o el bit de signo final. Como en el circuito de la figura 4.25 no está disponible al exterior el acarreo que se suma a los bits de signo, se utiliza el bit de signo del resultado.

Se produce un rebasamiento en las siguientes situaciones:

a) Si ambos operandos (aplicados a ambos sumadores) son positivos, lo cual se

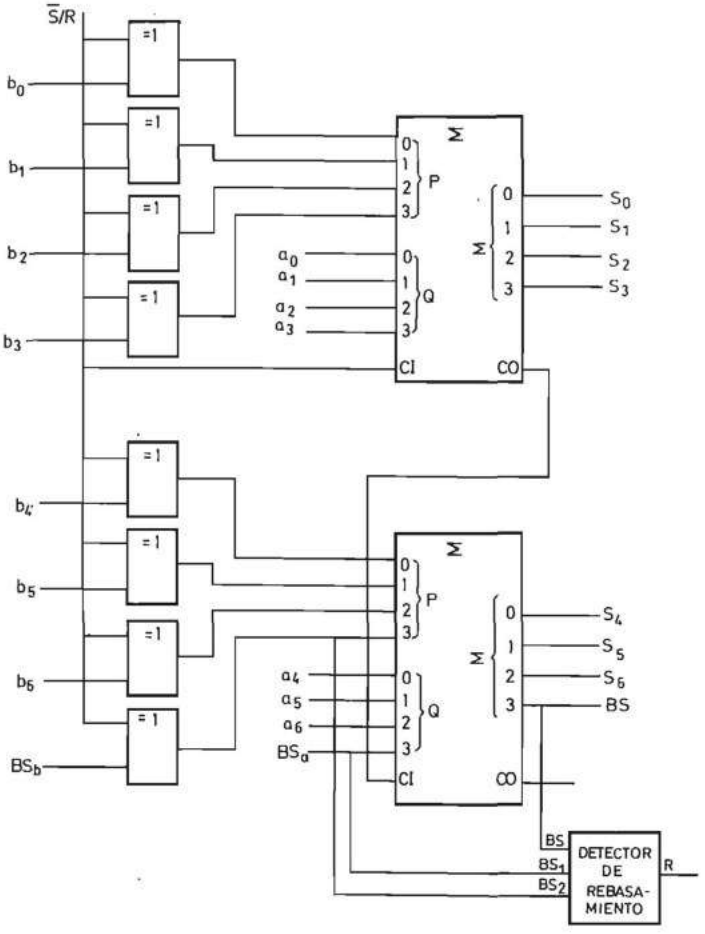


FIGURA 4.25.—**Circuito sumador-restador de números de siete bits y signo representados en el convenio del complemento a dos.**

detecta porque sus bits de signo son cero, y el bit de signo del resultado es un uno lógico.

b) Si ambos operandos son negativos, lo cual se detecta porque sus bits de signo son uno lógico, y el bit de signo del resultado es cero.

La ecuación lógica de la función R que toma el valor uno cuando se produce un rebasamiento es:

$$R = BS_1 \quad BS_2 \quad \overline{BS} + \overline{BS_1} \quad \overline{BS_2} \quad BS$$

que adecuadamente transformada resulta:

$$R = \overline{\overline{BS_1 \quad BS_2 \quad \overline{BS} + \overline{BS_1} \quad \overline{BS_2} \quad BS}} = \overline{\overline{BS_1 \quad BS_2 \quad \overline{BS}} \quad \overline{\overline{BS_1} \quad \overline{BS_2} \quad BS}}$$

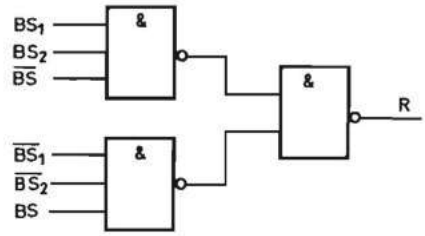


FIGURA 4.26.—**Esquema del detector de rebasamiento de la figura 4.25**

En la figura 4.26 se representa el circuito detector de rebasamiento realizado con puertas NO-Y.

4.2.3.3 Operación con números negativos mediante el convenio del complemento a uno. Al igual que en el convenio del complemento a 2^n, se estudia a continuación la resta del número b del número a, ambos de n bits (n — 1 bits de información y un bit de signo). Los casos que se pueden presentar son los mismos y seguidamente realizaremos su análisis:

1) a y b son ambos positivos.

Representando $-b$ por su complemento a $2^n - 1$, para restar b de a realizaremos la operación:

$$a - b \equiv a + 2^n - 1 - b = 2^n - 1 + a - b$$

Si $a > b$ el resultado será positivo y además se produce un acarreo.

Se tiene:

$$a - b \equiv 2^n + (a - b - 1)$$

El resultado obtenido es igual a la diferencia menos una unidad, lo cual se corrige sumando el acarreo obtenido.

Ejemplo:

$$\begin{array}{rl} & B.S. \\ a = 25_{10} \equiv & 0\ \ 1\ 1\ 0\ 0\ 1_2 \\ b = 18_{10} \equiv & 0\ \ 1\ 0\ 0\ 1\ 0_2 \end{array}$$

$-b$ se representa:

$$\begin{array}{c} B.S. \\ 1\ \ 0\ 1\ 1\ 0\ 1_2 \end{array}$$

La diferencia 25 — 18 resulta:

$$\begin{array}{rr} & B.S. \\ 25 & 0\ \ 1\ 1\ 0\ 0\ 1 \\ -18 & +1\ \ 0\ 1\ 1\ 0\ 1 \\ \hline 7 & 1\ 0\ \ 0\ 0\ 1\ 1\ 0 \\ & +\ \text{└}\!\!\rightarrow 1 \\ \hline & 0\ \ 0\ 0\ 1\ 1\ 1 \end{array}$$

Sumando el acarreo al resultado obtenido se tiene:

$$\begin{array}{rc} & B.S. \\ 7 \equiv & 0\ \ 0\,0\,1\,1\,1 \end{array}$$

Si $a < b$, el resultado será negativo y queda expresado correctamente mediante el complemento a $2^n - 1$.

$$a - b \equiv a + 2^n - 1 - b = 2^n - 1 - (b - a)$$

Ejemplo:

$$\begin{array}{rcl} & B.S. & \qquad\qquad & B.S. \\ a = 18_{10} \equiv & 0\ \ 1\,0\,0\,1\,0 & b = 25_{10} \equiv & 0\ \ 1\,1\,0\,0\,1 \end{array}$$

$-b$ se representa:

$$\begin{array}{c} B.S. \\ 1\ \ 0\,0\,1\,1\,0 \end{array}$$

La diferencia 18 − 25 resulta:

$$\begin{array}{rc} & B.S. \\ 18 & 0\ \ 1\,0\,0\,1\,0 \\ -25 & +1\ \ 0\,0\,1\,1\,0 \\ \hline -7 & 1\ \ 1\,1\,0\,0\,0 \end{array}$$

El lector puede comprobar que el resultado es el complemento a $2^n - 1$ de $7_{10} \equiv 000111_2$.

2) a y b son ambos negativos.

La diferencia entre $-a$ y $-b$ resulta:

$$\begin{aligned} (-a) - (-b) &= (2^n - 1 - a) + [2^n - 1 - (2^n - 1 - b)] = \\ &= 2^n - 1 - a + 2^n - 1 - 2^n + 1 + b = 2^n - 1 - (a - b) \end{aligned}$$

Al igual que en el caso anterior, si el resultado es positivo, se produce un acarreo que debe de sumarse al bit menos significativo y, si el resultado es negativo, queda representado mediante su complemento a $2^n - 1$.

Ejemplos:

$$\begin{array}{rc} & B.S. \\ -a = -20_{10} \equiv & 1\ \ 0\,1\,0\,1\,1 \\ -b = -24_{10} \equiv & 1\ \ 0\,0\,1\,1\,1 \end{array}$$

$$\begin{array}{rc} & B.S. \\ -(-b) = 24_{10} = & 0\ \ 1\,1\,0\,0\,0 \end{array}$$

$$\begin{array}{rc} & B.S. \\ -20 & 1\ \ 0\,1\,0\,1\,1 \\ -\ -24 & +0\ \ 1\,1\,0\,0\,0 \\ \hline 4 & 1\,0\ \ 0\,0\,0\,1\,1 \\ & +\ \text{└}\!\!\rightarrow 1 \\ \hline & 0\ \ 0\,0\,1\,0\,0 \end{array}$$

Resulta:

$$\begin{array}{r} \\ 4_{10} \equiv \end{array} \begin{array}{c} B.S. \\ 0\ \ 0\ 0\ 1\ 0\ 0_2 \end{array}$$

Si se tiene:

$$\begin{array}{rl} & B.S. \\ -a = -24_{10} \equiv & 1\ \ 0\ 0\ 1\ 1\ 1 \\ -b = -20_{10} \equiv & 1\ \ 0\ 1\ 0\ 1\ 1 \end{array}$$

$$\begin{array}{rl} & B.S. \\ -(-b) = 20_{10} = & 0\ \ 1\ 0\ 1\ 0\ 0 \end{array}$$

$$\begin{array}{rcl} & & B.S. \\ -24 & & 1\ \ 0\ 0\ 1\ 1\ 1 \\ -\ -20 & & +0\ \ 1\ 0\ 1\ 0\ 0 \\ \hline -\ 4 & & 1\ \ 1\ 1\ 0\ 1\ 1 \end{array}$$

El resultado obtenido es el complemento con respecto a 2^n-1 de $4_{10} \equiv 000100$.

3) a es positivo y b negativo.

La diferencia $a - (-b)$ da como resultado un número positivo y no se produce acarreo.

En efecto:

$$a - (-b) = a + 2^n - 1 - (2^n - 1 - b) = a + b$$

Ejemplo:

$$\begin{array}{rl} & B.S. \\ a = 14_{10} \equiv & 0\ \ 0\ 1\ 1\ 1\ 0 \\ -b = -17_{10} \equiv & 1\ \ 0\ 1\ 1\ 1\ 0 \\ -(-b) = 17_{10} \equiv & 0\ \ 1\ 0\ 0\ 0\ 1 \end{array}$$

$$\begin{array}{rcl} 14 & & 0\ \ 0\ 1\ 1\ 1\ 0 \\ -\ -17 & & +0\ \ 1\ 0\ 0\ 0\ 1 \\ \hline 31 & & 0\ \ 1\ 1\ 1\ 1\ 1 \end{array}$$

4) a es negativo y b positivo.

La diferencia $-a\ -b$ da como resultado un número negativo al cual ha de sumársele el acarreo producido para que quede expresado correctamente, tal como se indica seguidamente:

$$\begin{aligned} -a\ -b &= (2^n - 1 - a) + 2^n - 1 - b = \\ &= 2^n + 2^n - 2 - (a + b) \end{aligned}$$

Ejemplo:

$$\begin{array}{rl} & B.S. \\ a = 12_{10} \equiv & 0\ \ 0\ 1\ 1\ 0\ 0 \\ b = 17_{10} \equiv & 0\ \ 1\ 0\ 0\ 0\ 1 \\ -a = -12_{10} \equiv & 1\ \ 1\ 0\ 0\ 1\ 1 \\ -b = -17_{10} \equiv & 1\ \ 0\ 1\ 1\ 1\ 0 \end{array}$$

-12	$1\ 1\ 0\ 0\ 1\ 1$
$-\ 17$	$+1\ 0\ 1\ 1\ 1\ 0$
-29	$1\ 1\ 0\ 0\ 0\ 1$
	$+\ \longrightarrow 1$
	$1\ 0\ 0\ 0\ 1\ 0$

Queda también demostrado que, mediante el convenio del complemento a $2^n - 1$, la resta se reduce a una suma.

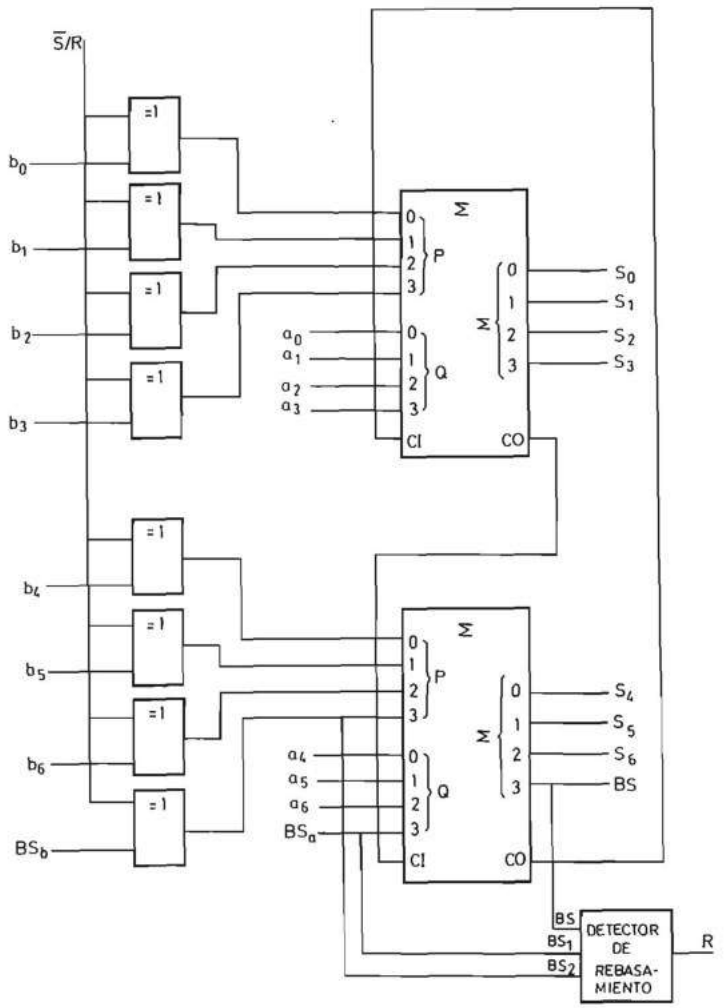


FIGURA 4.27.—**Circuito sumador-restador de números de siete bits y signo representados en el convenio del complemento a uno.**

A continuación se diseña un sumador-restador de 7 bits y signo. Se supone que los números negativos se presentan a su entrada representados en el convenio del complemento a 2^n—1. Para su realización se utilizan dos cuádruples sumadores como el representado en la figura 4.19.

La única diferencia entre este sumador-restador y el estudiado en el apartado anterior (figura 4.25) es que la entrada de acarreo del primer sumador debe conectarse a la salida de acarreo para que se realice su suma con los números *a* y *b*.

En la figura 4.27 se representa el esquema completo del sumador-restador en el convenio del complemento a 2^n—1.

4.2.4 Unidades aritméticas y lógicas

Tal como se estudia en el tomo II, en numerosos sistemas digitales es necesario realizar diferentes operaciones aritméticas y lógicas con números binarios. Esta necesidad indujo a los fabricantes de circuitos integrados a realizar en escala de integración media bloques funcionales de cuatro bits que, mediante unas entradas de selección, permitiesen realizar varias operaciones aritméticas y lógicas con números de la citada dimensión. Estos circuitos reciben el nombre de unidades aritméticas y lógicas (UAL) [Arithmetic Logic Unit (ALU)] y existen en diferentes versiones.

En la figura 4.28 se representa el símbolo lógico de una UAL típica, que se caracteriza por:

a) Poseer una tabla de verdad que depende del juego de operaciones aritméticas y lógicas que realice. La representada en la figura 4.28 opera en paralelo con combinaciones de cuatro bits y posee cinco terminales de modo de operación

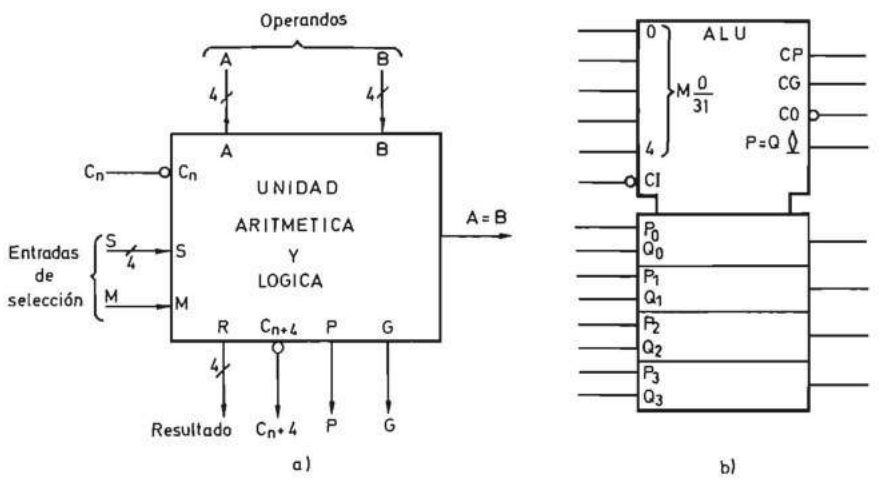


FIGURA 4.28.—**Unidad aritmética y lógica de cuatro bits: *a*) símbolo lógico no normalizado; *b*) símbolo lógico normalizado.**

que permiten seleccionar un total de 32 operaciones diferentes de las que dieciséis son aritméticas y otras tantas son lógicas. La distinción entre uno y otro grupo se realiza mediante el nivel cero o uno de la variable 4 en la figura 4.28*b* (*M* en la 4.28*a*).

En la tabla 4.3 se indican las operaciones aritméticas y lógicas que suelen realizar una unidad aritmética y lógica.

Operaciones aritméticas	*Operaciones lógicas*
R = A	R = $\overline{A}$
R = A + B	R = $\overline{A + B}$
R = $\overline{A}$B	R = $\overline{A}$B
R = −1	R = 0
R = A MAS A$\overline{B}$	R = $\overline{AB}$
R = (A + B) MAS A$\overline{B}$	R = $\overline{B}$
R = A MENOS B MENOS 1	R = A $\oplus$ B
R = A$\overline{B}$ MENOS 1	R = A$\overline{B}$
R = A MAS AB	R = $\overline{A}$ + B
R = A MAS B	R = $\overline{A \oplus B}$
R = (A + $\overline{B}$) MAS AB	R = B
R = AB MENOS 1	R = AB
R = A MAS A desplaz.	R = 1
R = (A + B) MAS A	R = A + $\overline{B}$
R = (A + $\overline{B}$) MAS A	R = A + B
R = A MENOS 1	R = A

TABLA 4.3

b) Realizar la generación del acarreo en paralelo para lograr la máxima velocidad de operación. En su interior existe un circuito generador de acarreos como los representados en las figuras 4.9 y 4.13. La UAL (ALU) de la figura 4.28*b* genera *CO* y además *CG* y *CP*.

Las operaciones lógicas se realizan entre los bits del mismo peso. Por ejemplo, si la operación seleccionada es $R = \overline{A + B}$ y $A = 0101$ y B = 0110, los bits del resultado se obtienen realizando la suma lógica de los bits del mismo peso de A y B e invirtiendo el resultado:

$$\overline{0 + 0} = 1$$
$$\overline{1 + 1} = 0$$
$$\overline{0 + 1} = 0$$
$$\overline{1 + 0} = 0$$

de donde resulta:

$$R = 1000$$

c) Generar una salida de igualdad $A = B$ ($P = Q$ en la figura 4.28*b*) de colector abierto.

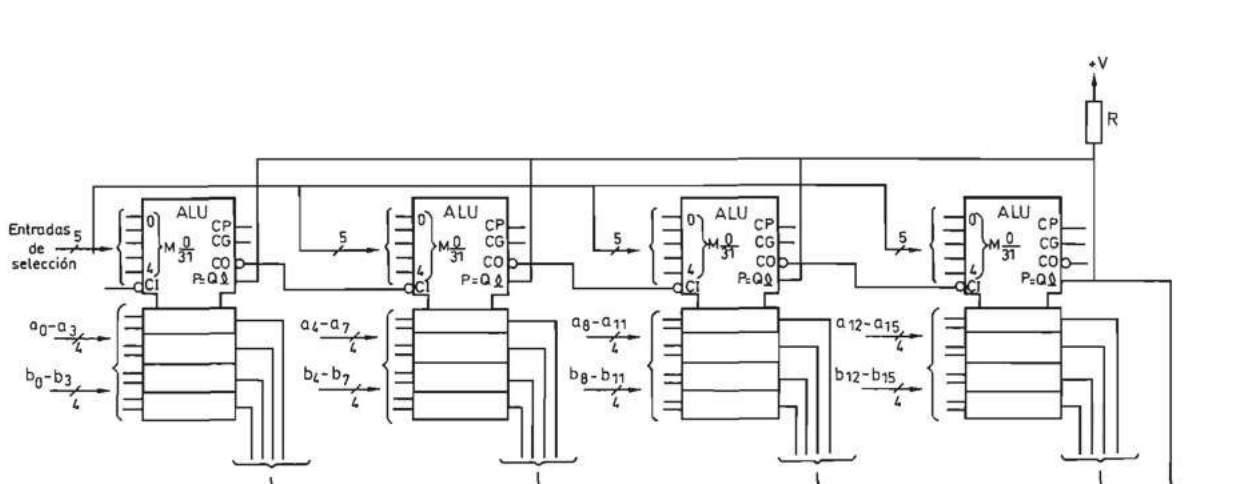


FIGURA 4.29.—**Esquema de una unidad aritmética y lógica de dieciséis bits realizada con cuatro bloques funcionales como el de la figura 4.28, conectados con propagación del acarreo en serie.**

Mediante la utilización de la salida de acarreo *CO* se realiza el acoplamiento de las UAL (ALU) con propagación del acarreo en serie. En la figura 4.29 se representa el esquema de una unidad aritmética y lógica de dieciséis bits realizada con cuatro bloques funcionales de cuatro bits cada uno, conectando la salida del acarreo de cada bloque *CO* a la entrada del siguiente. Las entradas de modo de operación de todos los bloques se conectan entre sí. Las salidas *P* y *G* no se utilizan.

La salida de comparación del conjunto se obtiene uniendo todos los terminales $P=Q$ entre sí, y a través de una resistencia al terminal positivo de la alimentación (uno en lógica positiva). Sólo si se produce la igualdad de todos los bits correspondientes de las combinaciones *A* y *B*, todas las salidas $A = B$ adoptan el nivel uno lógico y la salida común se encuentra en este nivel. Si alguna de las salidas $A = = B$ se encuentra en estado cero debido a que no son iguales las combinaciones *a* y *b* presentes en sus entradas, la salida $a = b$ adopta el nivel cero.

Mediante la utilización de los terminales de salida *CG* y *CP*, y un circuito generador de acarreos se realiza el acoplamiento de las UAL (ALU) con propagación del acarreo en paralelo. En la figura 4.30 se representa el esquema de una unidad aritmética y lógica de dieciséis bits con generación del acarreo en paralelo realizada con cuatro bloques funcionales y un generador de acarreos como el representado en la figura 4.9. Las salidas *P* y *G* de cada unidad aritmética y lógica de cuatro bits se conectan al generador de acarreos y éste genera en paralelo los bits de acarreo que se conectan a la entrada de cada una de ellas. Las salidas *CO* no se utilizan.

En el circuito de la figura 4.30, se obtiene la salida de comparación igual que en el de la figura 4.29.

4.2.5 Multiplicación binaria

La multiplicación de dos números binarios de un bit a_0 y b_0 cumple la tabla de verdad indicada en la tabla 4.4 que coincide con el producto lógico, es decir:

$$P = a_0 b_0$$

tal como se representa gráficamente en la figura 4.31.

El algoritmo de multiplicación utilizado en el sistema binario natural coincide con el del sistema decimal. Se multiplica el multiplicando por el bit

b_0	a_0	P
0	0	0
0	1	0
1	0	0
1	1	1

TABLA 4.4

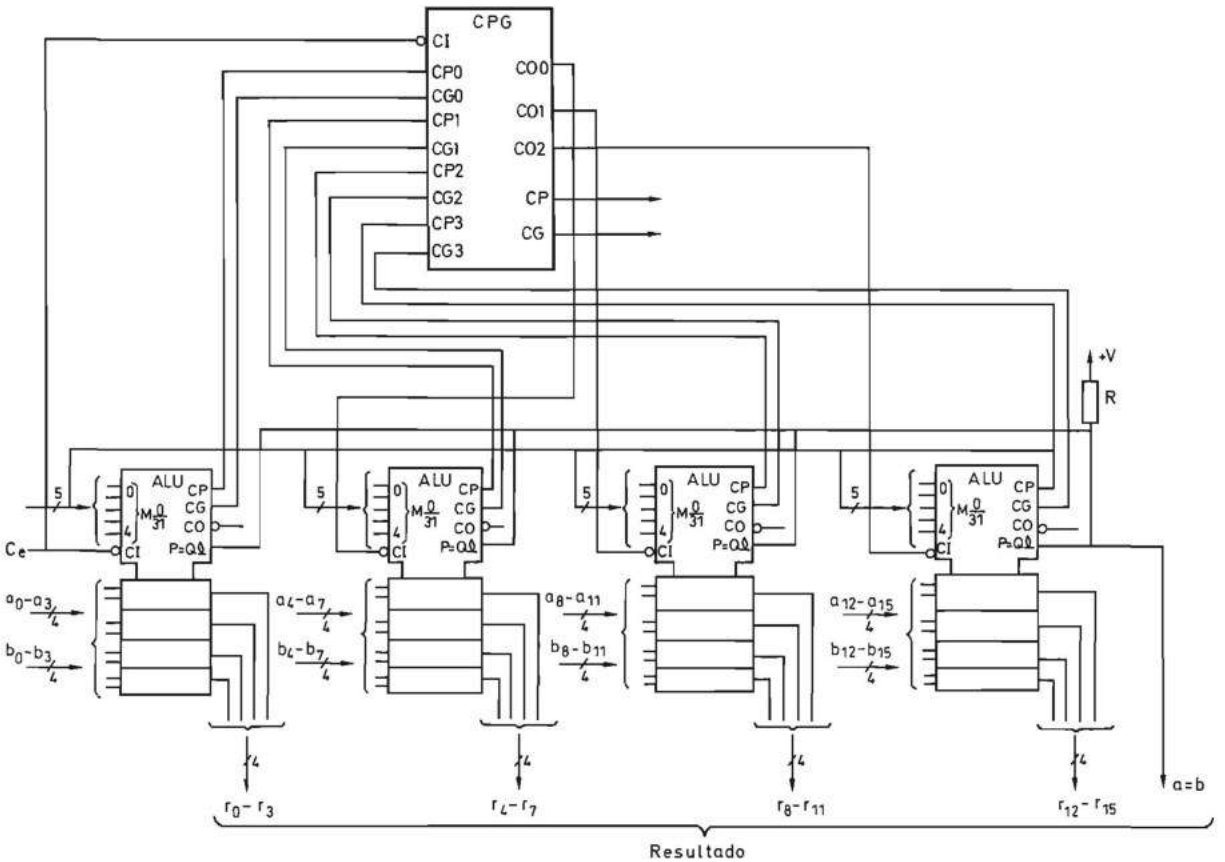


FIGURA 4.30.—**Esquema de una unidad aritmética y lógica de dieciséis bits realizada con cuatro bloques funcionales como el de la figura 4.28 y un circuito generador de acarreos en paralelo.**

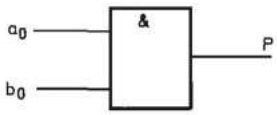


FIGURA 4.31.—**Multiplicador de un bit.**

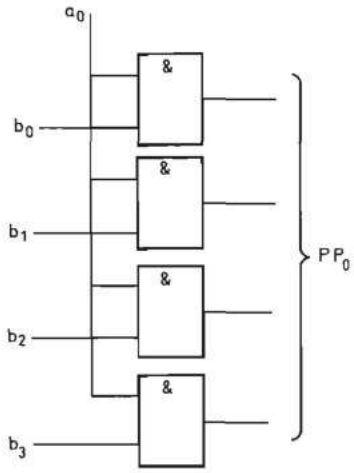


FIGURA 4.32.—**Generador del producto parcial PP_0.**

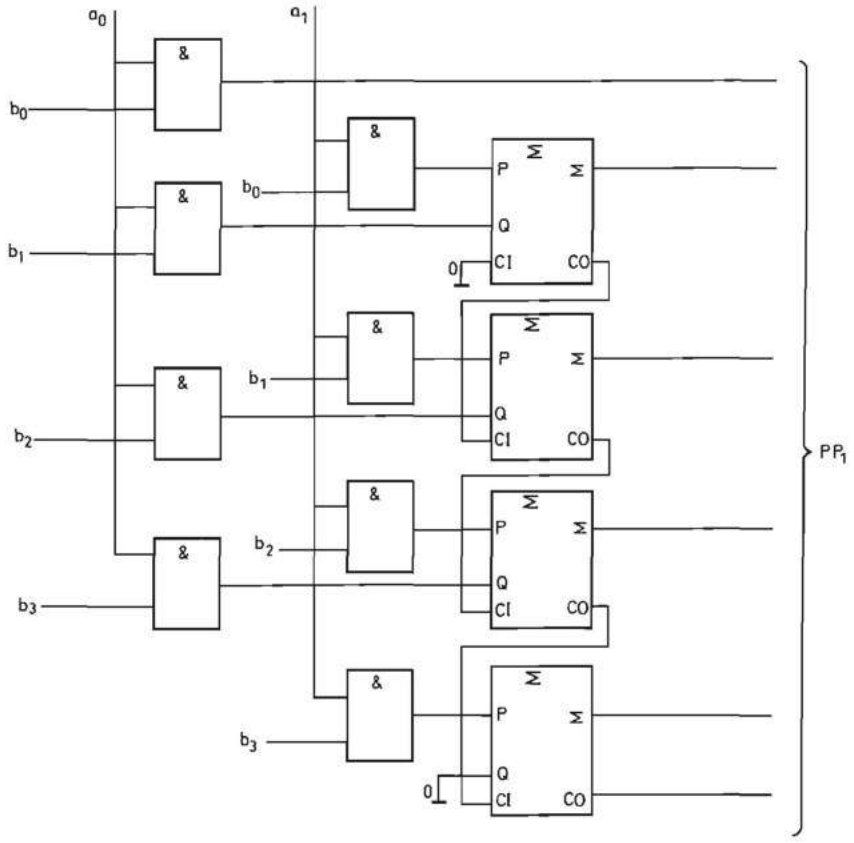


FIGURA 4.33.—**Generador de los productos parciales PP_0 y PP_1.**

menos significativo del multiplicador. Al resultado obtenido se le suma aritméticamente el producto, desplazado una posición del segundo bit del multiplicador por el multiplicando, y así sucesivamente.

Trataremos de realizar este algoritmo de forma iterativa utilizando puertas Y para realizar cada uno de los productos parciales y sumadores totales para realizar las sumas parciales.

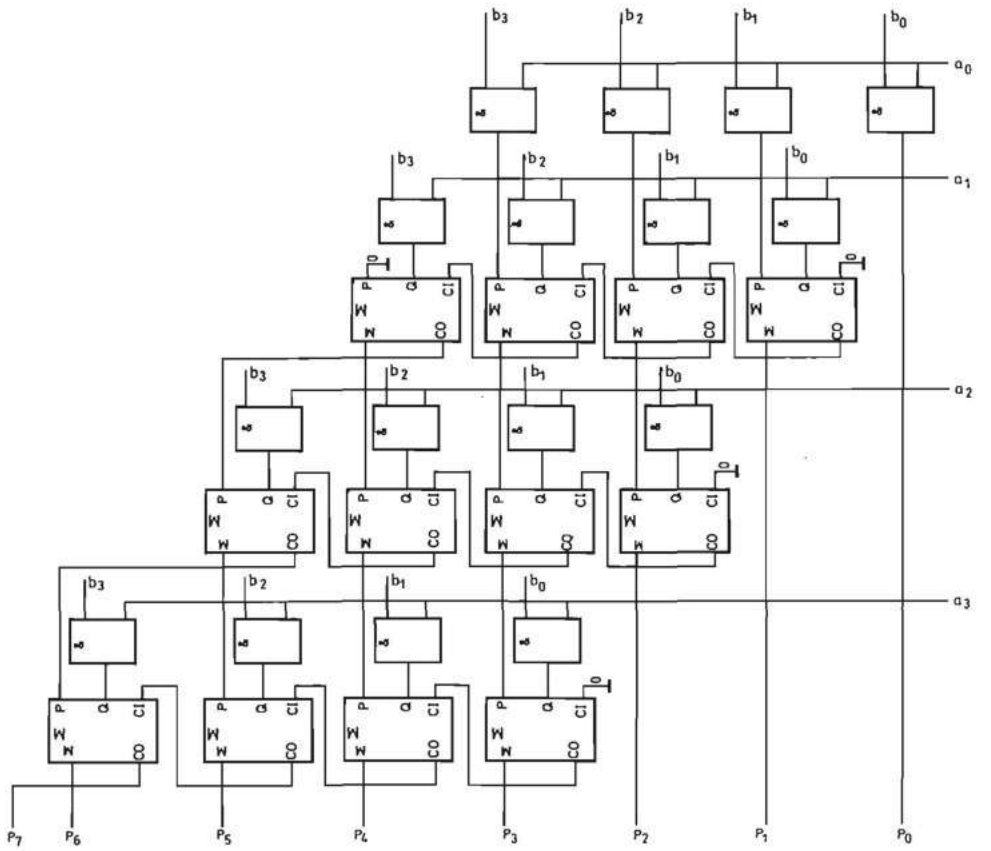


FIGURA 4.34.—**Multiplicador de números de cuatro bits.**

Aclararemos lo expuesto con números de 4 bits:

$$B = b_3 \quad b_2 \quad b_1 \quad b_0 \quad \text{Multiplicando}$$
$$A = a_3 \quad a_2 \quad a_1 \quad a_0 \quad \text{Multiplicador}$$

La multiplicación de a_0 por B se realiza mediante cuatro puertas Y, tal como se indica en la figura 4.32, obteniéndose el primer producto parcial PP_0. Este producto PP_0 hay que sumarlo con el segundo producto parcial $a_1 \times B$ desplazado un bit. Para ello se pueden utilizar cuatro sumadores totales y otras cuatro puertas Y,

tal como se indica en la figura 4.33. Utilizando este procedimiento de forma iterativa se obtiene el esquema de la figura 4.34

Si, después de las cuatro puertas Y que realizan el primer producto parcial, se añaden cuatro sumadores totales cuyas entradas se unen a las salidas de las puertas Y y al nivel lógico cero, se obtiene el esquema del multiplicador de la figura 4.35 en el cual se observa que el bloque formado por una puerta Y y un sumador total (fig. 4.36) se utiliza como elemento básico para realizar el multiplicador.

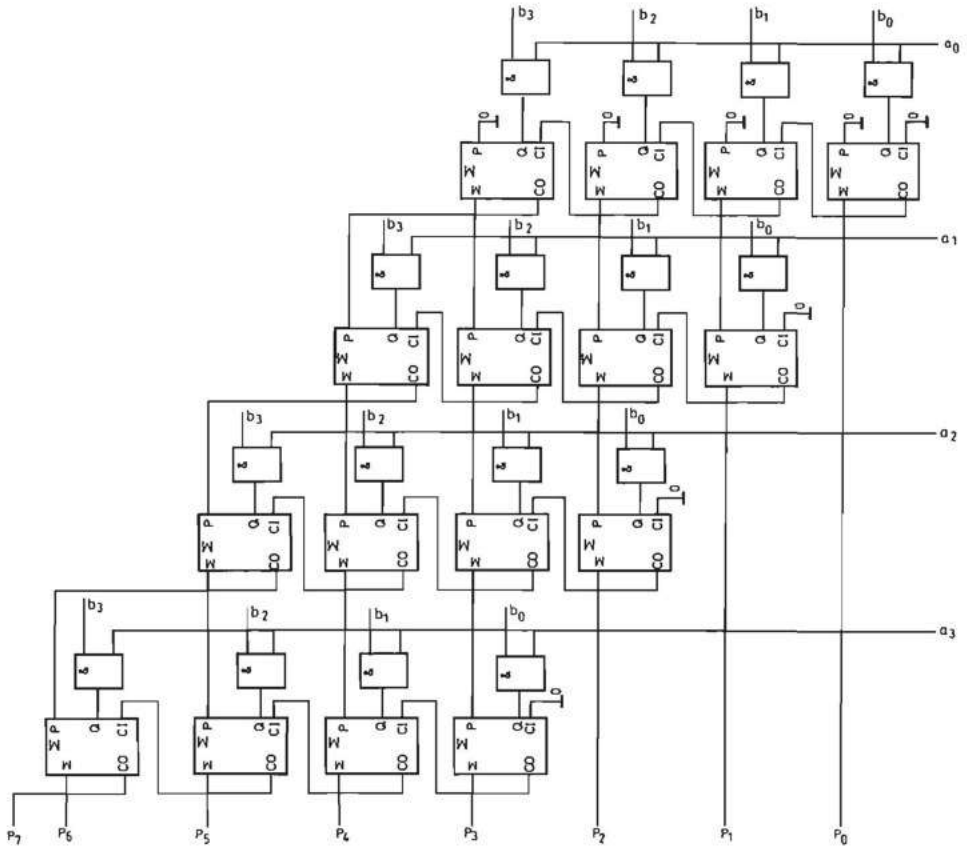


FIGURA 4.35.—**Multiplicador de números de cuatro bits.**

El lector puede observar que en la figura 4.35 se realiza la suma con propagación del acarreo en serie. Se podría también realizar con propagación anticipada, lo cual complicaría el esquema.

Por ser la multiplicación una operación que se realiza con frecuencia en múltiples sistemas digitales, han sido realizados bloques funcionales multiplicadores de números positivos o negativos que, acoplados adecuadamente, permiten la realización de un multiplicador de cualquier número de bits.

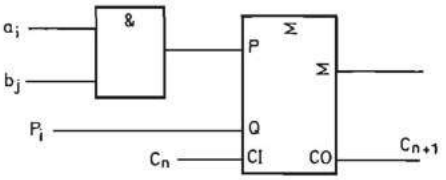


FIGURA 4.36.—**Elemento básico de un multiplicador.**

La multiplicación se puede realizar también mediante un único sumador y un conjunto de registros, controlados por un sistema secuencial síncrono. El sistema así obtenido constituye un procesador digital secuencial síncrono cuyo estudio se realiza en el tomo II.

4.3 OPERACIONES ARITMETICAS EN LOS CODIGOS DECIMALES CODIFICADOS EN BINARIO (BCD)

Como se indicó en el capítulo 1, los códigos decimales codificados en binario [Binary Coded Decimal (BCD)] han sido muy utilizados en el diseño de pequeños sistemas digitales antes de que el progreso de la microelectrónica permitiese la integración de los microprocesadores. La utilización de técnicas de diseño de circuitos integrados digitales totalmente a medida (full custom) y semimedida (semicustom) vuelve a hacer interesante su utilización en aquellos casos en que el número de circuitos a fabricar lo permite.

Por todo ello en sucesivos apartados se estudia el diseño de los circuitos sumador y restador en el código BCD natural. El sumador y el restador en códigos autocomplementarios como el BCD exceso tres ha perdido interés porque su única ventaja es que su circuito es un poco más sencillo que el BCD natural, y por ello no se incluye su estudio.

4.3.1 Operaciones en el código decimal codificado en binario natural (BCD natural)

4.3.1.1 Suma en el código decimal codificado en binario natural (BCD natural). Estudiaremos el circuito sumador de dos digitos y el acarreo procedente de una etapa anterior. Mediante acoplamiento de este circuito con otros idénticos se puede realizar la suma de números en BCD natural de cualquier longitud.

Para realizar este diseño utilizaremos el circuito cuádruple sumador total con propagación del acarreo en serie.

Si utilizamos un cuádruple sumador total para realizar la suma de dos décadas y el acarreo procedente de una etapa anterior, se pueden obtener los resultados decimales y binarios indicados en las dos primeras columnas de la tabla 4.5. De estos resultados, los comprendidos del cero al nueve no necesitan corrección pues son equivalentes a las combinaciones del código BCD natural indicado en la tercera columna.

Por el contrario, los resultados del 10 al 19 no son correctos y se observa que,

añadiendo el número 6_{10} (0110_2) al resultado de la segunda columna, se obtiene el de la tercera. En la tercera columna se representa el bit de acarreo que se sumará a la etapa siguiente. Dicho bit de acarreo debe de tomar el valor uno para los resultados decimales 10 a 19.

Un ejemplo aclarará lo que acabamos de explicar. Por ejemplo, la suma de los números decimales 24_{10} y 17_{10} representados en BCD natural será:

	1		
24	0 0 1 0	0 1 0 0	
+ 17	+ 0 0 0 1	0 1 1 1	
41	0 1 0 0	1 0 1 1	
	+	0 1 1 0	Corrección
	0 1 0 0	1 0 0 0 1	

y el resultado obtenido es:

$$0100 \quad 0001 \equiv 41_{10}$$

Resultado Suma de dos dígitos	*Acarreo*	*Suma a corregir*				*Acarreo Decimal*	*Suma Decimal*			
	C_4	S_4	S_3	S_2	S_1	C_D	S_{D8}	S_{D4}	S_{D2}	S_{D1}
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

TABLA 4.5

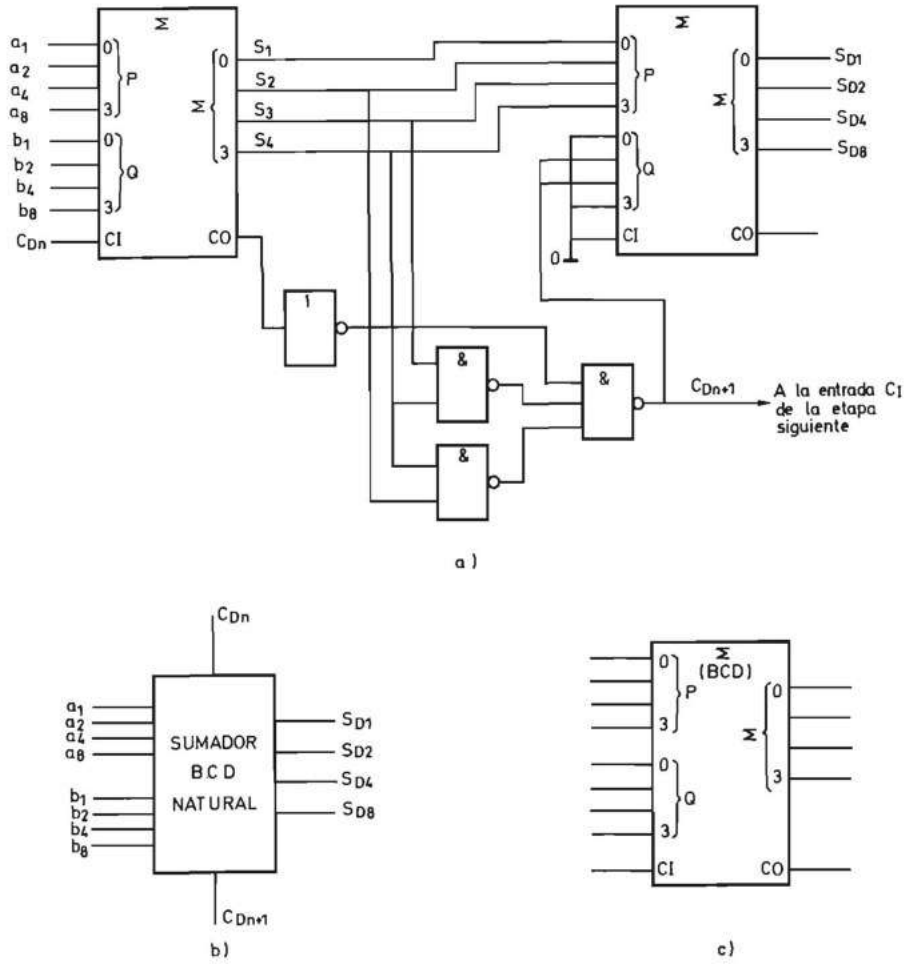


FIGURA 4.37.—**Sumador de dos dígitos en el código BCD natural: *a*) esquema del circuito; *b*) símbolo lógico no normalizado; *c*) símbolo lógico normalizado.**

En la figura 4.37*a* se representa el circuito completo de un sumador de dos dígitos en el código BCD natural. El primer sumador total obtiene los resultados de la segunda columna de la tabla 4.5.

El acarreo que hay que aplicar a la etapa siguiente debe de ser un uno lógico si lo es la salida de acarreo del primer sumador, o bien, si el resultado obtenido en éste está comprendido entre 10 y 15. Realizando la tabla de Karnaugh de productos canónicos del acarreo representada en la figura 4.38 y agrupando los unos al máximo se obtiene la expresión mínima del acarreo decimal C_D que es:

$$C_D = S_2S_4 + S_3S_4 + C_4$$

que, realizada con puertas NO-Y, resulta:

$$C_D = \overline{\overline{S_2 S_4} \quad \overline{S_3 S_4} \quad \overline{C_4}}$$

La corrección de la suma se realiza mediante un segundo cuádruple sumador total a cuyas entradas A_1 a A_4 se conectan las salidas correspondientes del primer sumador. Sus entradas B_1 y B_4 se conectan a un nivel de cero lógico y las entradas B_2 y B_3 a la señal de acarreo C_{Dn+1}. De esta forma se suma $0_{10}(0000)$ al resultado del primer sumador si está comprendido entre 0 y 9, y $6_{10}(0110)$ si está comprendido entre 10 y 19.

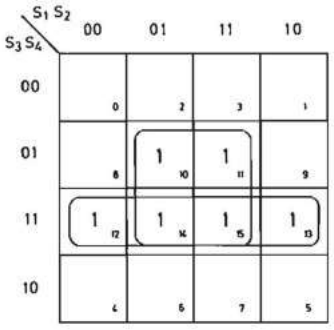


FIGURA 4.38

En la figura 4.37*b* y *c* se representa el símbolo lógico del sumador de dos dígitos en BCD natural, en sus versiones no normalizada y normalizada, respectivamente.

Mediante la conexión de n bloques como el que acabamos de describir, se realiza la suma de un número BCD natural de n dígitos.

En la figura 4.39 se representa el esquema correspondiente.

4.3.1.2 Resta en el código decimal codificado en binario natural (BCD natural). La resta en el sistema decimal se puede realizar de una forma similar al sistema de base dos, sumando al minuendo el complemento a 10^n (complemento a 10) o el complemento a $10^n - 1$ (complemento a nueve) del sustraendo. En los sistemas BCD se utiliza corrientemente el convenio del complemento a nueve porque permite reducir la complejidad del sistema.

La operación de la resta del número b del a se representa por:

$$a - b \equiv a + 10^n - 1 - b = 10^n - 1 + (a - b) = 10^n + (a - b - 1)$$

De esta ecuación se deduce que, si el resultado es negativo, no se produce acarreo y el número obtenido queda representado mediante su complemento a nueve. Si el resultado es positivo, se produce un acarreo que ha de sumarse a aquél para obtener la suma correcta.

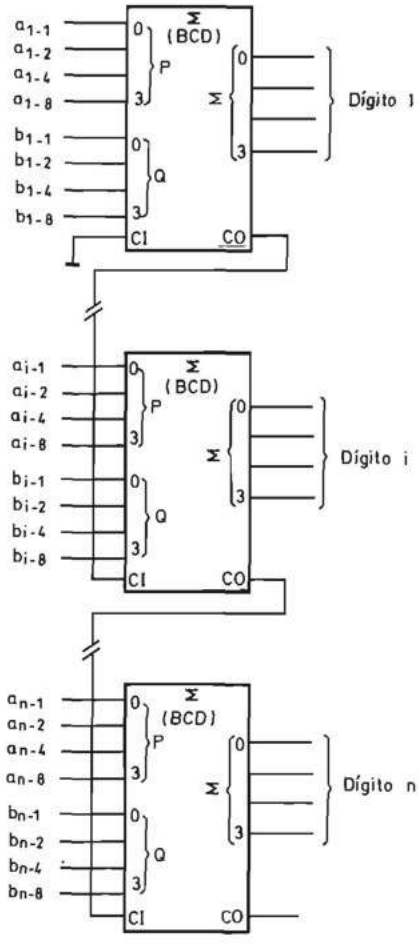


FIGURA 4.39.—**Sumador de números de *n* dígitos en BCD natural realizado con el bloque funcional de la figura 4.37.**

Un ejemplo aclarará lo que acabamos de expresar. Sean los números decimales 24 y 17. Su representación en BCD natural es:

$$24_{10} \equiv (0010 \quad 0100) \quad \text{BCD natural}$$
$$17_{10} \equiv (0001 \quad 0111) \quad \text{BCD natural}$$

Los números negativos correspondientes se representarán en el convenio del complemento a nueve:

$$-24_{10} \equiv 99 - 24 = 75$$
$$-17_{10} \equiv 99 - 17 = 82$$

La representación en BCD natural será:

$$-24_{10} \equiv -0010 \quad 0100 \equiv 1001 \quad 1001 - 0010 \quad 0100 = 0111 \quad 0101$$
$$-17_{10} \equiv -0001 \quad 0111 \equiv 1001 \quad 1001 - 0001 \quad 0111 = 1000 \quad 0010$$
$$-24_{10} \equiv (0111 \quad 0101) \quad \text{BCD natural}$$
$$-17_{10} \equiv (1000 \quad 0010) \quad \text{BCD natural}$$

para lo cual hemos restado cada dígito en BCD natural de

$$1001 \equiv 9_{10}$$

Como ejemplo realizaremos la resta 24 − 17. Primeramente en el sistema decimal.

$$\begin{array}{r} 24 \\ -17 \\ \hline 7 \end{array} \qquad \begin{array}{r} 24 \\ +82 \\ \hline 1\ 06 \\ +\ \hookrightarrow 1 \\ \hline 7 \end{array}$$

En el código BCD natural tendremos:

$$\begin{array}{r} 24 \\ -17 \\ \hline 7 \end{array} \qquad \begin{array}{rr} 0010 & 0100 \\ +1000 & 0010 \\ \hline 1010 & 0110 \\ +0110 & \\ \hline 1\ 0000 & 0110 \\ +\ \hookrightarrow & 1 \\ \hline 0000 & 0111 \equiv 7_{10} \end{array} \quad \text{Corrección}$$

Realicemos ahora la resta 17 − 24

$$\begin{array}{r} 17 \\ -24 \\ \hline -7 \end{array} \qquad \begin{array}{r} 17 \\ +75 \\ \hline 92 \end{array} = \text{Complemento a nueve de 7.}$$

En el código BCD natural tendremos:

$$\begin{array}{r} 17 \\ -24 \\ \hline -7 \end{array} \qquad \begin{array}{rr} 1 & \\ 0001 & 0111 \\ 0111 & 0101 \\ \hline 1001 & 1100 \\ + & 0110 \\ \hline 1001 & 1\ 0010 \end{array} \quad \text{Corrección}$$

El resultado obtenido 1001 0010 es el complemento a 1001 1001 de 0000 0111 $\equiv 7_{10}$.

En la figura 4.40 se representa el esquema de bloques de un restador en el código BCD natural de n dígitos. El circuito sumador en BCD natural ha sido diseñado en el apartado 4.3.1.1 y representado en la figura 4.37. Debemos diseñar el circuito generador del complemento a nueve. Su tabla de verdad se representa en la tabla 4.6 en la que observamos que se trata de una multifunción que puede ser sintetizada mediante los métodos estudiados en el capítulo 3. Pero, por otra parte, el complemento a nueve de un dígito a se puede calcular invirtiendo el número y sumándole $1010 \equiv 10_{10}$. En efecto:

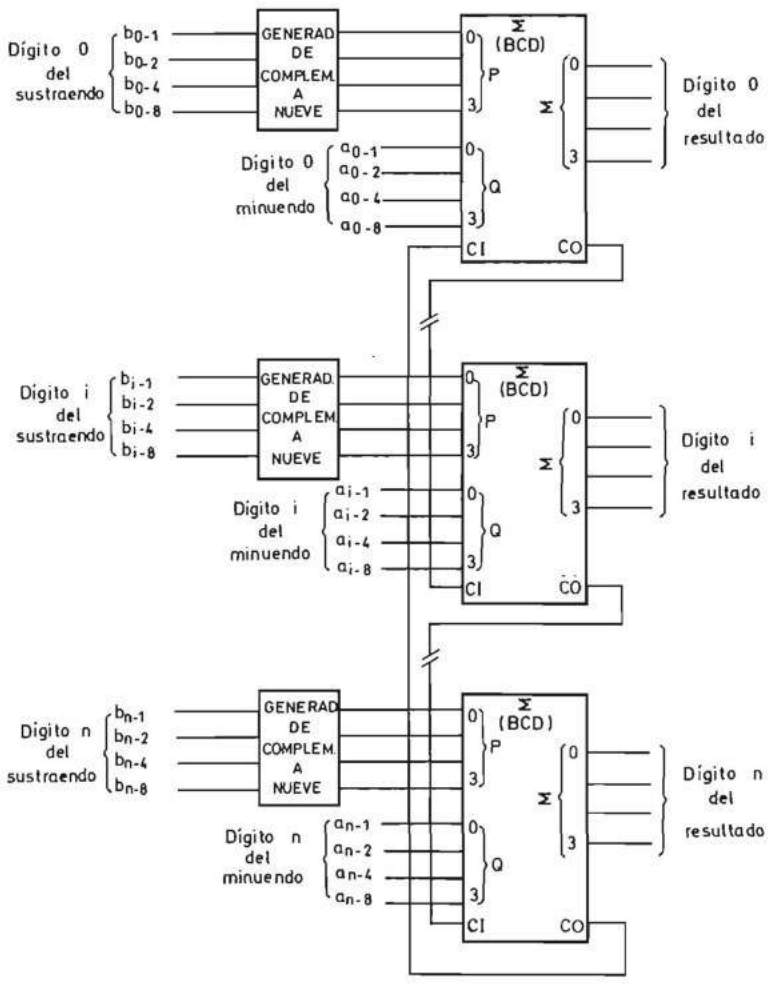


FIGURA 4.40.—**Restador de números de n dígitos en BCD natural representados en el convenio del complemento a nueve.**

				Complemento a nueve			
a_4	a_3	a_2	a_1	d_4	d_3	d_2	d_1
0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

TABLA 4.6.—**Tabla de verdad del generador del complemento a nueve en el código BCD natural.**

$$9 - a \equiv 2^4 - 1 - a + 10 = 2^4 + 9 - a$$

produciéndose un acarreo que no es tenido en cuenta.

En la figura 4.41*a* se representa el circuito que realiza el complemento a nueve. Los cuatro inversores no son necesarios si están disponibles los bits $\overline{a_1}$ a $\overline{a_4}$. En la figura 4.41*b* se indica el símbolo lógico del generador del complemento a nueve.

Para operar simultáneamente con números positivos y negativos es conveniente utilizar el convenio del complemento a nueve junto con un bit de signo, que puede ser uno si el número es negativo y cero si es positivo.

Realizaremos seguidamente algunos ejemplos de operación con números negativos utilizando ambos convenios.

Sean los números:

	B.S.		
$33_{10} \equiv$	0	0011	0011
$42_{10} \equiv$	0	0100	0010

Los mismos números negativos serán:

	B.S.		
$-33_{10} \equiv$	1	0110	0110
$-42_{10} \equiv$	1	0101	0111

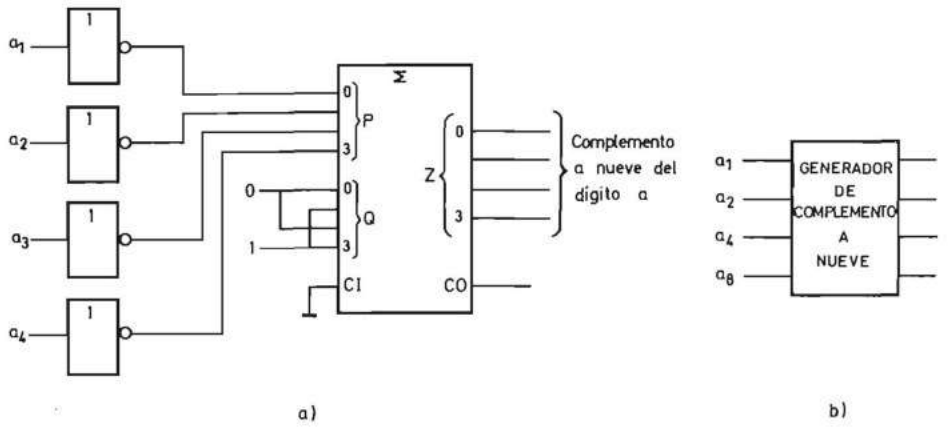


FIGURA 4.41.—**Generador de complemento a nueve: *a*) esquema del circuito; *b*) símbolo lógico.**

La operación 33 − 42 será:

	B.S.		
		1	
33	0	0011	0011
− 42	+ 1	0101	0111
− 9	1	1001	1010
	+		0110 Corrección
	1	1001	10000

El resultado, por lo tanto, es:

$$1 \quad 1001 \quad 0000 \equiv -9_{10}$$

El bit de signo 1 indica que el resultado es negativo y, por tanto, los dígitos están representados mediante su complemento a nueve. En efecto, el complemento a nueve de 1001 0000 es 0000 1001 equivalente al 9 decimal.

La operación 42 − 33 será:

	B.S.			
42	0	0100	0010	
−33	+1	0110	0110	
9	1	1010	1000	
	+1	0110		Corrección
	1 0 1	0000	1000	
+			→1	
	0	0000	1001	

El resultado es:

$$\begin{matrix} B.S. \\ 0 \quad 0000 \quad 1001 \equiv 9_{10} \end{matrix}$$

En la figura 4.42 se representa un sistema sumador-restador en el código BCD natural.

El número *a*, que es el minuendo en la operación resta, se aplica directamente a las entradas de los circuitos sumadores en BCD natural. El número *b* se conecta, por un lado, a las entradas de un circuito generador del complemento a nueve que acabamos de estudiar y, además, directamente a las entradas de un cuádruple multiplexor de dos canales. Asignando a la variable de selección *E* de los multiplexores el estado cero o uno lógicos, se consigue que se presenten a las entradas del sumador el número *b* o su complemento a nueve respectivamente, por lo que de esta forma se selecciona la realización de la operación suma o resta respectivamente. Además, en el caso de la operación resta, se debe de invertir el signo del número *b*, lo cual se realiza mediante una puerta O-exclusiva controlada por la misma variable.

Al igual que en el circuito sumador-restador binario natural en el convenio del complemento a uno (figura 4.27), el acarreo de la suma de los bits de signo se conecta a la entrada de acarreo del primer sumador.

Es interesante también detectar la situación en que el resultado rebase la capacidad máxima del sumador. El rebasamiento se produce en los siguientes casos:

a) Aplicando a los sumadores dos números positivos, es decir, cuyo bit de signo es cero en ambos, se produce un uno lógico en el acarreo C_{Dn+1} del último sumador.

b) Aplicando a los dos sumadores dos números negativos, es decir, cuyo bit de signo es uno en ambos, se produce un cero lógico en el acarreo C_{Dn+1} del último sumador con lo cual el resultado sería positivo, lo cual es erróneo.

Por tanto, denominando BS_a y BS_o a los bits de signo de ambos números, la ecuación lógica de la variable rebasamiento *R* resulta:

$$R = \overline{BS_a} \quad \overline{BS_b} \quad C_{Dn+1} + BS_a \quad BS_b \quad \bar{C}_{Dn+1}$$

que adecuadamente transformada queda finalmente:

$$R = \overline{\overline{\overline{BS_a} \quad \overline{BS_b} C_{Dn+1}} \quad \overline{BS_a \, BS_b \, \bar{C}_{Dn+1}}}$$

En la figura 4.43 se representa el esquema del circuito detector de rebasamiento realizado con puertas NO-Y.

4.4 FORMATOS DE REPRESENTACION DE LOS NUMEROS FRACCIONARIOS

En los apartados anteriores se estudiaron las operaciones aritméticas con números codificados en el sistema binario natural o en los códigos decimales codificados en binario natural o en exceso tres considerándolos enteros.

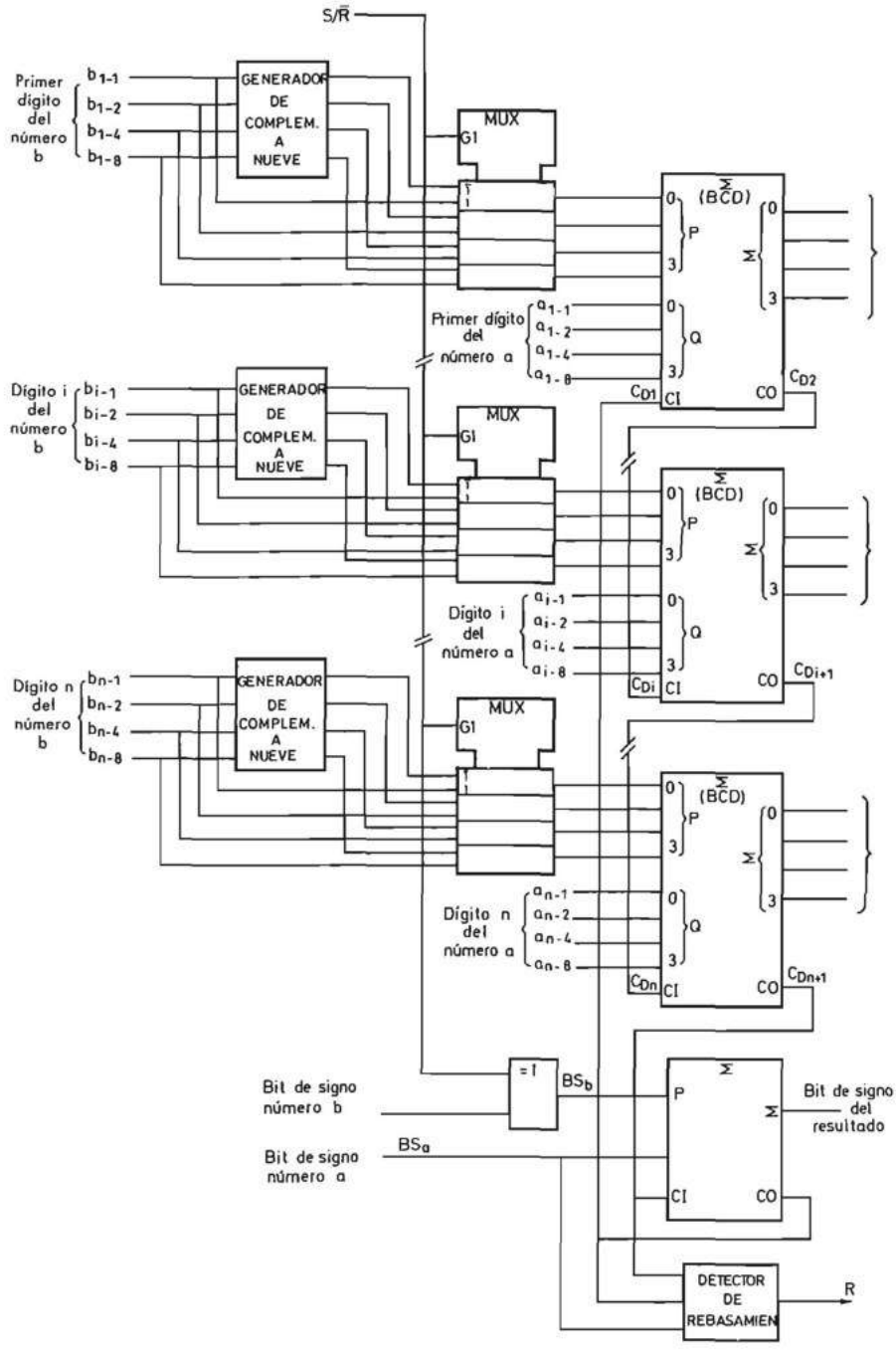


FIGURA 4.42.—**Sumador-restador de números de *n* dígitos en BCD natural con circuito detector de rebasamiento.**

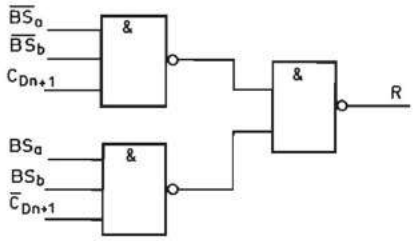


FIGURA 4.43.—**Esquema del circuito de rebasamiento de la figura 4.42.**

Las operaciones con números fraccionarios introducen un nuevo concepto, que es el de la coma que separa la parte entera de la parte fraccionaria.

Estudiaremos las formas o sistemas de representar números fraccionarios y la realización de operaciones en cada uno de ellos.

4.4.1 Representación de los números fraccionarios en coma fija

En esta forma de representación se asigna, tal como su nombre indica, una posición fija a la coma. Por ejemplo, si se opera con números binarios de ocho bits, la coma puede situarse de forma arbitraria en cualquiera de las posiciones, pero una vez elegida no se modifica.

La principal ventaja de este método de representación es que los algoritmos de realización de las diferentes operaciones son los mismos que los estudiados para los números enteros en los apartados anteriores.

Por ejemplo, sean los siguientes números de ocho bits con dos decimales: 011011,11 y 100001,01. La suma de ambos números es:

$$\begin{array}{r} 0\,1\,1\,0\,1\,1,\,1\,1 \\ +\;1\,0\,0\,0\,0\,1,\,0\,1 \\ \hline 1\,1\,1\,1\,0\,1,\,0\,0 \end{array}$$

y se observa que la operación se realiza como si los números fuesen enteros.

El principal inconveniente de la representación en coma fija es que no se aprovecha la capacidad de los operadores aritméticos. En efecto, si se suponen, al igual que antes, números de ocho bits con dos fraccionarios, el máximo número que puede representarse es 111111,11 y el mínimo distinto de cero es 0,01. Pero la capacidad de los operadores podría permitir operar el número máximo 11111111 y el mínimo 0,00000001.

4.4.2 Representación de los números fraccionarios en coma flotante

Este modo de representación evita el inconveniente antes citado en los números de coma fija, pero, tal como se comprueba seguidamente, los algoritmos de realización de las diferentes operaciones son más complejos.

La representación de los números en coma flotante en un sistema de numeración de base B se realiza mediante una mantisa m y un exponente e tal que:

$$n = m\,B^e$$

m: Número en coma fija que se denomina mantisa.
e: Número entero que se denomina exponente.
B: Base del sistema de numeración.

En el sistema de numeración decimal ($B = 10$) un ejemplo de número representado en el formato de coma flotante es el $22{,}5 \cdot 10^3$. Pero este número se puede representar de muy diversas maneras:

$$22{,}5 \cdot 10^3 = 0{,}0225 \cdot 10^6 = 225.000 \cdot 10^{-1} = \ldots$$

que se diferencian por la situación de la coma en la mantisa y el valor del exponente.

Pero, en la práctica, es necesario elegir una de todas las formas posibles para realizar las operaciones en ella. Las dos más utilizadas son:

— Situación de la coma inmediatamente antes de la cifra más significativa distinta de cero.
En el ejemplo anterior es: $0{,}225 \cdot 10^5$

— Situación de la coma inmediatamente después de la cifra más significativa distinta de cero.
En el ejemplo anterior es: $2{,}25 \cdot 10^4$

De lo expuesto se deduce que el paso de una forma de representación a otra se realiza desplazando la coma tantos lugares como sea necesario hacia la izquierda o hacia la derecha y aumentando o disminuyendo el valor del exponente en tantas unidades como desplazamientos se hayan efectuado.

En el sistema de numeración binario natural se pueden representar también los números en coma flotante.

Por ejemplo:

$$0{,}011 \cdot 2^{0011} = 1{,}1 \cdot 2^{0001} = 110 \cdot 2^{-0001} = \ldots$$

Al igual que en el sistema de numeración decimal, las dos expresiones normalizadas más utilizadas son:

$$0{,}11 \cdot 2^{0010}$$
$$1{,}1 \cdot 2^{0001}$$

Diversas formas (denominadas formatos) de representación de los números binarios en coma flotante han sido desarrolladas por distintos fabricantes de procesadores digitales.

Dichos formatos definen:

— El número total de bits utilizados.
— La forma de representación de números positivos y negativos de la mantisa.

FIGURA 4.44

— La forma de representación de números positivos y negativos del exponente.
— El número de bits de la mantisa y del exponente.
— El orden correlativo entre la mantisa y el exponente. El más utilizado se indica en la figura 4.44.

Para la mantisa se escoge una de las formas de representación de los números indicada en el apartado 4.2.3.1: bit de signo y valor absoluto, bit signo y complemento a uno o bit de signo y complemento a dos.

Para escoger la forma de representar los números en el exponente es necesario tener en cuenta que para realizar una suma en coma flotante es necesario previamente:

— Comparar los exponentes.
— Incrementar el menor en tantas unidades como lo diferencian del mayor.

Para comparar fácilmente los exponentes es conveniente que los números negativos se representen mediante combinaciones binarias menores que las de los números positivos. Esto se consigue asignando el valor uno al bit de signo de los exponentes positivos y representando los exponentes negativos mediante el complemento a dos de su correspondiente positivo. Por ejemplo suponiendo cuatro bits para representar el exponente, el exponente +3 se representa mediante la combinación 1011 y el exponente −3 mediante su complemento a dos, es decir 0101.

Esta forma de representación recibe el nombre de polarizada (en inglés «biased») y el código así obtenido binario natural desplazado (en inglés «binary offset).

En la tabla 4.7 se indica la representación de los exponentes de cuatro bits en binario natural desplazado junto al binario natural en complemento a dos.

De acuerdo con este criterio, si se dedican seis bits a la mantisa codificada en binario natural y complemento a dos, y cuatro al exponente, el número $-0{,}11 \times 2^{0010}$ se representa por:

SM

1	1 0 1 0	0 1 0 0 0 0

El «Institute of Electrical & Electronic Engineers» (IEEE) aprobó en 1985 el formato normalizado IEEE 754. Este formato tiene tres versiones de 32, 64 y 128 bits respectivamente, la primera de las cuales se representa en la figura 4.45 y está formada por el bit de signo de la mantisa, el exponente de 8 bits y la mantisa de 23 bits.

Exponente decimal	*Codificación del exponente en binario desplazado*				*Binario natural complemento a dos*			
−8	0	0	0	0	1	0	0	0
−7	0	0	0	1	1	0	0	1
−6	0	0	1	0	1	0	1	0
−5	0	0	1	1	1	0	1	1
−4	0	1	0	0	1	1	0	0
−3	0	1	0	1	1	1	0	1
−2	0	1	1	0	1	1	1	0
−1	0	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	1
2	1	0	1	0	0	0	1	0
3	1	0	1	1	0	0	1	1
4	1	1	0	0	0	1	0	0
5	1	1	0	1	0	1	0	1
6	1	1	1	0	0	1	1	0
7	1	1	1	1	0	1	1	1

TABLA 4.7

La mantisa es un número representado en valor absoluto cuyo bit de signo es cero si el número es positivo y uno si es negativo. La elección del valor absoluto en lugar del complemento a dos para representar la mantisa es debido a que así simplifican la multiplicación y la división que son operaciones muy utilizadas en coma flotante. Además la mantisa está normalizada de tal manera que la coma queda situada a la derecha de la cifra distinta de cero más significativa. Dado que esta cifra es un uno, no es necesario incluirla en el campo de la mantisa, que, por lo tanto, contiene los 23 bits que se encuentran a la derecha de la coma.

El exponente es un número representado en binario natural desplazado que, por lo tanto, puede variar entre −128 y +127.

La complejidad de los algoritmos de operación en coma flotante hace que su sistema físico sea asimismo muy complejo. Por ello, hasta que no se alcanzó el nivel de muy gran escala de integración (VLSI), las operaciones en coma flotante se realizaron mediante un circuito operador de coma fija y un sistema secuencial sín-

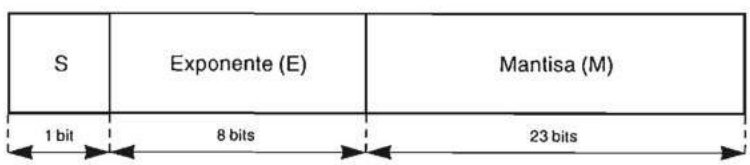


FIGURA 4.45.—**Formato IEEE 754-1985 de 32 bits.**

crono que constituyen conjuntamente un procesador digital secuencial síncrono que ejecuta las operaciones en coma flotante mediante una secuencia de operaciones básicas. En la actualidad los procesadores digitales suelen tener circuitos operadores en coma flotante que reciben la denominación de coprocesadores. Su análisis se sale fuera de los límites de este libro.

PROBLEMAS

1. Se tienen tres números de dos dígitos (8 bits) en el código BCD exceso tres. Diseñar un circuito que realice la suma de los dos mayores de los tres, utilizando al máximo los circuitos de escala de integración media.
2. ¿Cuál es el número de bits necesarios para sumar 17 y 19 con un bit de signo, representando estos números en el sistema binario natural?
3. Realizar la resta 47 — 39 en el sistema binario natural con un bit de signo:
   a) Representando los números negativos mediante el convenio del complemento a uno.
   b) Representando los números negativos mediante el convenio del complemento a dos.
4. Diseñar un circuito que realice la suma del número A de cuatro bits (a_1 a a_4) con el mayor de los dos números $B(b_1$ a $b_4)$ y $C(c_1$ a $c_4)$. Todos los números están codificados en el sistema binario natural. Si los números B y C son iguales, el resultado ha de ser igual al número A. Utilizar al máximo circuitos de escala de integración media.
5. Realizar las siguientes operaciones en el código BCD natural representando los números negativos mediante el complemento a nueve junto con un bit de signo. Indíquese en cada caso la capacidad que ha de tener el sumador para que no se produzca rebasamiento.
   a) 49 — 36.
   b) 54 + 72.
   c) −281 + 514.
   d) −58 — 63.
6. Diseñar un sumador-restador de 8 bits y un bit de signo con propagación del acarreo en paralelo. Utilícense al máximo los circuitos de escala de integración media.

Capítulo

5

Tecnologías de realización de los circuitos digitales

5.1 INTRODUCCION

Los métodos de diseño de sistemas digitales hasta ahora estudiados conducen a la realización de un circuito de conmutación.

Para ello se han de utilizar dispositivos físicos que posean dos estados bien diferenciados a los cuales se les podrá asignar los valores cero y uno lógicos.

El primer elemento físico utilizado ha sido el relé con uno o más contactos y en general se adoptó el convenio de asignar el valor lógico uno al contacto cerrado y el valor lógico cero al contacto abierto.

Ha sido el desarrollo espectacular de la electrónica en los últimos años lo que ha permitido la realización de sistemas digitales cada vez más complejos y con un consumo de potencia cada vez más reducido.

En este estudio de la tecnología de los sistemas digitales se hará referencia exclusivamente a los circuitos electrónicos. Los estados diferenciados de los elementos electrónicos se identifican mediante dos tensiones distintas. El criterio de asignación de los estados uno y cero lógicos es totalmente arbitrario. Si se asigna el valor 1 a la tensión más alta y el valor 0 a la tensión más baja, la lógica utilizada se llama positiva y si, por el contrario, se asigna el valor 0 a la tensión más alta y el valor 1 a la tensión más baja, la lógica recibe el nombre de negativa.

A lo largo de este capítulo se estudian los distintos dispositivos semiconductores utilizados en la realización de los sistemas digitales y sus características más importantes.

5.2 CARACTERISTICAS GENERALES DE LOS CIRCUITOS DIGITALES

Las características generales de toda puerta lógica, independientemente de la función que la misma realice, son las siguientes:

a) Cargabilidad de salida (fan-out). Es el máximo número de puertas básicas que pueden ser gobernadas por una sola puerta.

b) Cargabilidad de entrada (fan-in). Es el máximo número de entradas que puede tener una puerta lógica.

c) Tensión de umbral. Se refiere a las puertas lógicas y es por definición la tensión en que la puerta comienza a cambiar de estado lógico. Existen dos tensiones de umbral, una correspondiente al estado lógico de entrada cero y otra al estado lógico de entrada uno. Se definen ambas tensiones en los puntos en que la pendiente de la curva de transferencia es igual a -1.

La curva de transferencia de una puerta lógica representa gráficamente la relación entre las tensiones de entrada y salida de ella. En la figura 5.1 se indica la curva de transferencia de una puerta inversora en la que se pueden observar ambas tensiones de umbral $V_{u,0}$ y $V_{u,1}$.

d) Margen de ruido (noise margin). Es, por definición, la variación de tensión admisible a la entrada de un elemento lógico (de duración superior al tiempo de propagación medio del elemento considerado) sin que la salida del mismo cambie de estado.

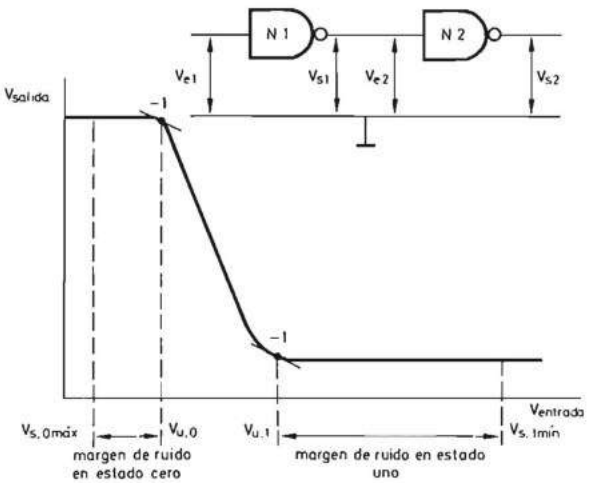


FIGURA 5.1.—**Curva de transferencia típica de un elemento lógico inversor.**

Existen dos márgenes de ruido, uno para el estado lógico de entrada cero y otro para el estado lógico de entrada uno.

La determinación de los márgenes de ruido se realiza sobre la curva de transferencia del elemento lógico correspondiente.

En la figura 5.1 se definen:

$V_{S,\,0\,max}$ = máxima tensión que puede aparecer a la salida de una puerta cuando su estado lógico es cero y está conectada al número máximo de puertas posible.

$V_{S,\,1\,min}$ = mínima tensión que puede aparecer a la salida de una puerta cuando su estado lógico es uno y está conectada al máximo número de puertas posible.

$V_{U,\,0}$ = tensión de umbral correspondiente al estado lógico de entrada cero.

$V_{U,\,1}$ = tensión de umbral correspondiente al estado lógico de entrada uno.

Normalmente una puerta se conecta a otra de características eléctricas idénticas tal como se indica en la figura 5.1 y, por tanto, la máxima tensión de entrada en estado cero de una puerta $V_{E,\,0\,máx.}$ es equivalente a la máxima tensión de salida en estado cero $V_{S,\,0\,máx.}$ de la puerta que está conectada a ella. Lo mismo se puede decir de la mínima tensión de entrada en estado uno de una puerta $V_{E,\,1\,mín.}$ Por consiguiente:

$$V_{E,0\,máx.} = V_{S,\,0\,máx.}$$
$$V_{E,\,1\,mín.} = V_{S,\,1\,mín.}$$

Se definen los márgenes de ruido de la forma siguiente:

Margen de ruido en estado lógico cero de la entrada: es la diferencia entre $V_{U,0}$ y $V_{E,0\,máx.}$

$$M_0 = V_{U,\,0} - V_{E,\,0\,máx.} = V_{U,0} - V_{S,0\,máx.}$$

Margen de ruido en estado lógico uno de la entrada: es la diferencia entre $V_{E,\,1\,mín.}$ y $V_{U,\,1}$

$$M_1 = V_{E,\,1\,mín.} - V_{U,1} = V_{S,\,1\,mín.} - V_{U,1}$$

Los valores de las cuatro tensiones que se acaban de definir dependen de la tecnología de la puerta considerada y son especificados por el fabricante de la misma.

e) Curva de inmunidad dinámica a los ruidos. Permite conocer el comportamiento de una puerta digital en régimen dinámico. Representa la relación entre la

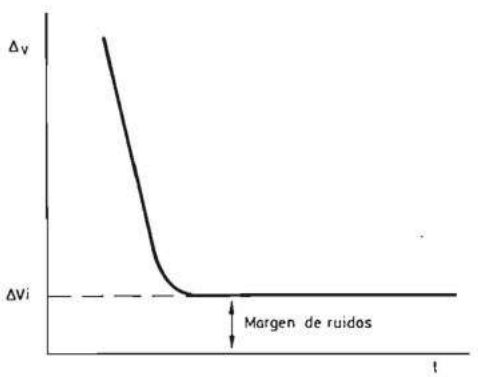


FIGURA 5.2.—**Curva de inmunidad dinámica al ruido.**

altura de los impulsos aplicados a su entrada y su duración mínima para hacer cambiar el estado lógico de la salida de la puerta (fig. 5.2). Existe una curva de este tipo para el estado uno de la entrada y otra para el estado cero. La tensión ΔV_i es el valor del margen de ruidos correspondiente de la puerta.

f) Tiempo de propagación medio (tpd). Se define como la media aritmética

entre los tiempos medios de propagación del cambio de estado de la entrada a la salida en los casos en que ésta pasa del estado 1 al 0, y viceversa

$$\text{tpd} = \frac{\text{tpdf} + \text{tpdr}}{2}$$

En la figura 5.3 se definen gráficamente los conceptos tpdf y tpdr aplicados al caso de una puerta que realiza la función lógica inversión.

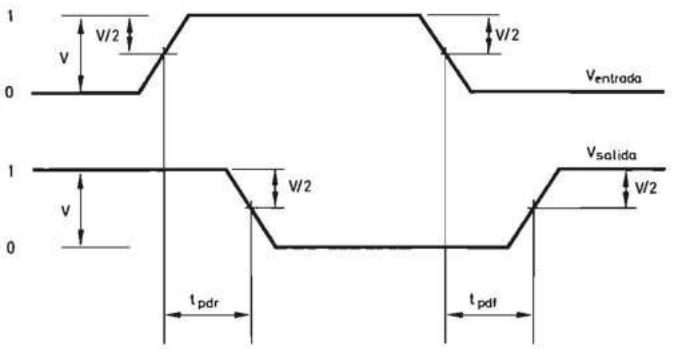


FIGURA 5.3.—**Relación temporal entre la señal de entrada y la de salida de un inversor.**

g) Potencia disipada: Se define para un ciclo de trabajo del 50 % y es la potencia disipada por una puerta lógica en esas condiciones.

h) Producto potencia disipada-tiempo de propagación: Como su nombre indica, es el producto de las dos características antes citadas. Constituye un factor de mérito de una determinada familia o tecnología de realización de los circuitos lógicos. Cuanto menor es el valor de este parámetro, mejor es la técnica de realización de la familia considerada.

5.3 CIRCUITOS DIGITALES CON DIODOS

El diodo semiconductor que presenta los estados de conducción y no conducción bien diferenciados, según el ánodo sea positivo o negativo, respectivamente, con respecto al cátodo, es un elemento idóneo para la realización de circuitos digitales.

En la figura 5.4 se representa una puerta elemental con diodos. Si cualquiera de las entradas se conecta al nivel de masa de la fuente que genera $+V_c$, la salida toma un nivel de tensión aproximadamente igual al de masa (incrementado en la caída de tensión directa en el diodo). Solamente si las tres entradas *a*, *b* y *c* están polarizadas al nivel $+V_C$, la salida estará también a dicho nivel. El lector puede deducir que esta puerta realiza la función Y en lógica positiva y la función O en lógica negativa.

La figura 5.5 representa otra puerta elemental con diodos. Se deja al lector que

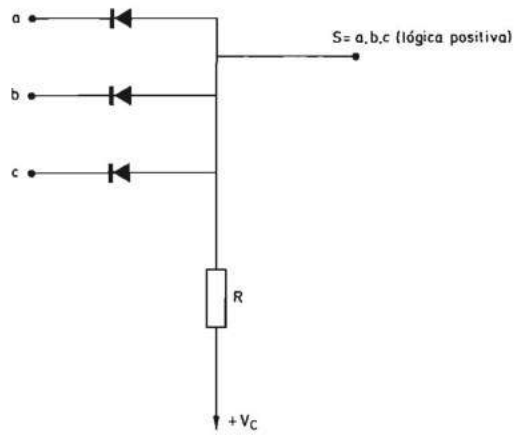


FIGURA 5.4.—**Puerta elemental con diodos que realiza la función Y en lógica positiva.**

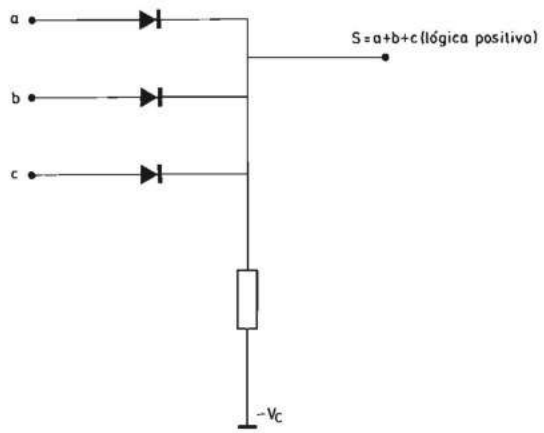


FIGURA 5.5.—**Puerta elemental con diodos que realiza la función O en lógica positiva.**

analice su comportamiento eléctrico y deduzca que esta puerta realiza la función O en lógica positiva y la función Y en lógica negativa.

Por ser el diodo un elemento pasivo en la práctica resulta imposible la conexión de puertas Y y puertas O realizadas con diodos. En efecto, analizando la figura 5.6, se deduce que la resistencia R_2 deberá ser muy elevada con respecto a la resistencia R_1 de la puerta Y_1 a fin de que cuando a_1, b_1 y c_1 están al nivel $+V$, el nivel de S_1 sea aproximadamente igual a $+V$. Pero, por otra parte, R_2 deberá ser mucho menor que la resistencia R_1 de la puerta Y_2 para que el nivel de S_2 sea aproximadamente igual al de masa cuando todas las entradas de la puerta O se encuentran en este nivel. Como no se pueden cumplir al mismo tiempo estas dos condiciones, se deduce la imposibilidad de tener puertas Y y puertas O normalizadas conectables entre sí.

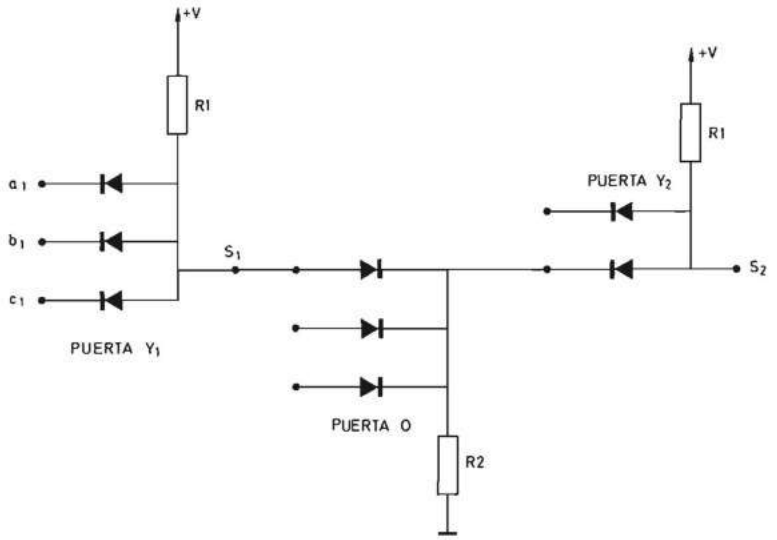


FIGURA 5.6.—**Circuito con puertas Y y puertas O realizadas con diodos.**

Más adelante se verá que estas puertas con diodos se utilizan combinadas con elementos activos.

Una aplicación típica de los diodos es la realización de convertidores de código como el que se indica en la figura 5.7, que realiza la conversión del sistema decimal al BCD natural.

5.4 CIRCUITOS DIGITALES CON TRANSISTORES

5.4.1 Características principales del transistor bipolar en conmutación

En conmutación, el transistor bipolar puede trabajar principalmente en saturación y en corriente (sin llegar a la saturación), lo cual, considerando I_B constante, depende de la resistencia de carga. En la figura 5.8 se representan ambas formas de funcionamiento. Si la recta de carga es R_1, el transistor se satura cuando la intensidad de base es I_{Bn} y, si la recta de carga es R_2, el transistor no se satura.

Dado que el funcionamiento sin llegar a la saturación se diferencia del de saturación fundamentalmente en la ausencia del tiempo de almacenamiento que más adelante definiremos, nos limitaremos a estudiar sucintamente el transistor bipolar en saturación. Para un estudio más profundo se remite al lector a obras especializadas en el tema [MILL 88].

En la figura 5.9 se representa un circuito que actúa de la forma mencionada.

Cuando al punto E se aplica la tensión de 0 V, el transistor bipolar está en corte (punto A de la fig. 5.8) y en su colector S aparece la tensión $+ V$ correspondien-

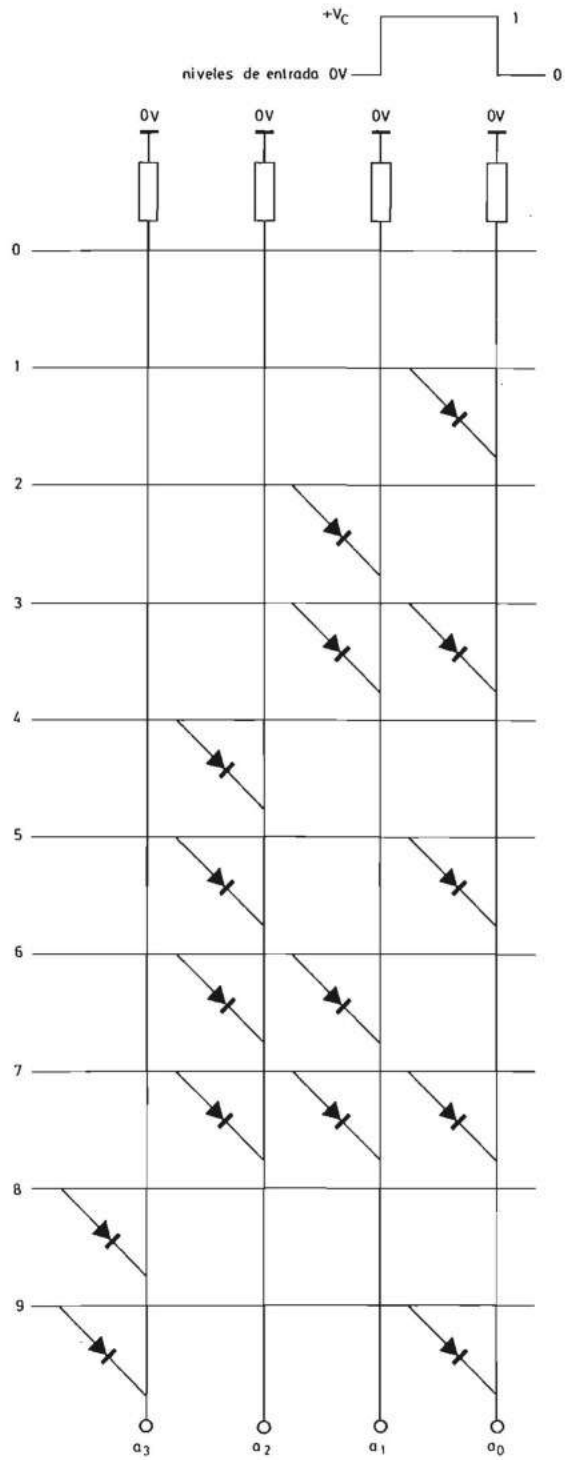


FIGURA 5.7.—**Convertidor del código decimal al BCD natural (lógica positiva) realizado con diodos.**

te al uno lógico. Cuando al punto E se aplica la tensión $+ V$ voltios, el transistor está saturado (punto B de la fig. 5.8) y la tensión en su colector es aproximadamente 0 V, correspondiente al cero lógico.

Además de las características estáticas de funcionamiento del transistor bipolar en cada uno de los dos estados citados de corte o saturación, es de gran importancia su comportamiento dinámico, es decir, la forma en que se produce el paso de un estado a otro.

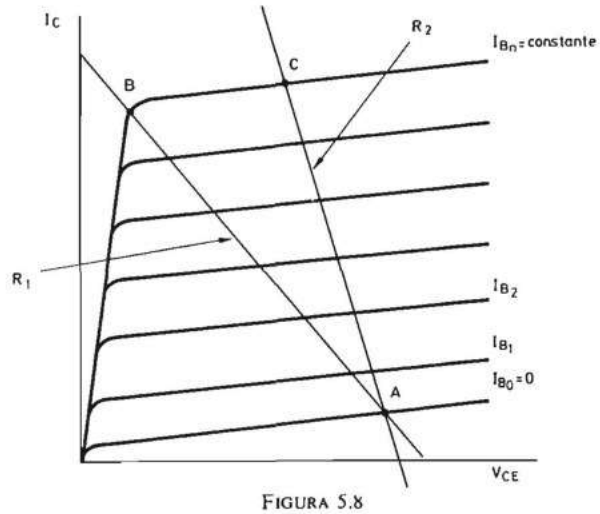


FIGURA 5.8

En la figura 5.10 se representa la respuesta de un transistor bipolar incorporado en el circuito de la figura 5.9 cuando a su entrada se aplica un impulso rectangular. Seguidamente se analiza dicha respuesta.

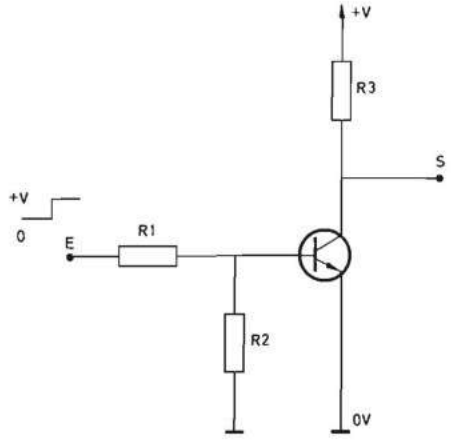


FIGURA 5.9.—**Circuito de conmutación básico.**

Desde el instante en que la tensión E pasa del valor 0 a $+ V$ hasta que la intensidad de colector alcanza el 90 % de su valor final I_{CS} y el transistor se satura, transcurre un tiempo t_1. Igualmente, desde que la tensión en E pasa del valor $+ V$ a 0 hasta que la intensidad de colector alcanza el 10 % de I_{CS} transcurre un tiempo t_2; t_1 y t_2 pueden ser denominados tiempos de conmutación del estado de corte al de saturación y viceversa, respectivamente.

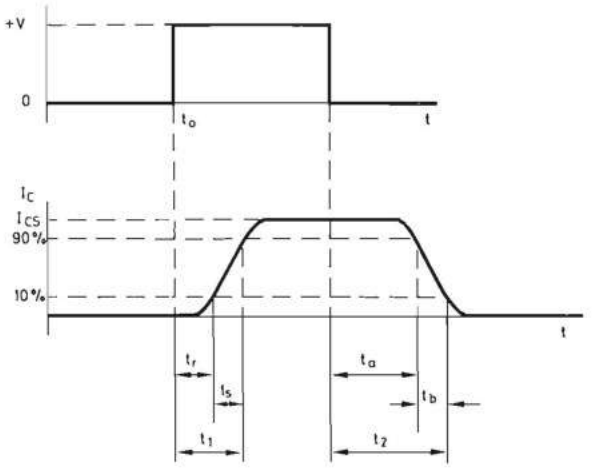


FIGURA 5.10.—**Respuesta de un transistor bipolar a un impulso rectangular.**

El tiempo de conmutación del estado de corte al de saturación t_1 puede dividirse en otros dos, t_r y t_s, tales que $t_1 = t_r + t_s$ (fig. 5.10). El tiempo t_r que tarda I_C en alcanzar el valor 0,1 I_{CS} depende de la resistencia interna de la fuente de tensión que proporciona el impulso de entrada y de las características físicas del transistor. El tiempo t_s, o tiempo de subida, es el que tarda I_C en pasar de 0,1 I_{CS} a 0,9 I_{CS} y depende de los parámetros del transistor y del factor de sobresaturación. Se define el factor de sobresaturación como la relación entre la intensidad de base que realmente circula y la que debe circular para que el transistor alcance la saturación y su expresión es

$$K_S = \frac{I}{I_{CS}/\beta}$$

siendo β el valor de la ganancia de intensidad en saturación del transistor. El valor mínimo de este factor debe ser la unidad para que el transistor alcance la saturación y cuanto mayor es su valor menor es t_S.

El tiempo de conmutación del estado de saturación al de corte t_2 puede a su vez dividirse en otros dos t_a y t_b (fig. 5.10). El tiempo t_a llamado de almacenamiento, es el necesario para la desaparición de los portadores de carga minoritarios almacenados en la base; t_a depende de la impedancia de salida y de las características

del transistor, y su valor aumenta al hacerlo el factor de sobresaturación. Se define t_a como el tiempo que transcurre desde que la tensión de entrada pasa de $+V$ a 0 voltios hasta que la intensidad de colector alcanza el valor 0,9 I_{CS}. El tiempo t_b llamado de bajada, es el que transcurre mientras I_C pasa del valor 0,9 I_{CS} a 0,1 I_{CS}. Su valor depende de las características propias del transistor.

Aunque una disminución del valor del factor de sobresaturación produce una disminución de t_a y una elevación de t_S es posible mantener éste muy bajo si las características del transistor son adecuadas. Por esto se deben evitar valores de K_S elevados (muy superiores a 1), con lo cual se logran circuitos de conmutación muy rápidos.

Los circuitos lógicos con transistores bipolares en saturación han sido realizados mediante diferentes técnicas de las cuales las que más se utilizan —por lo que las describiremos— son las de lógica resistencia-transistor (RTL), lógica diodo-transistor (DTL) y lógica transistor-transistor (TTL).

En un principio los circuitos lógicos han sido realizados mediante componentes discretos. La existencia de un número reducido de funciones lógicas elementales ha permitido la fabricación en serie primeramente de circuitos bloque y posteriormente de circuitos integrados monolíticos. La utilización de estos componentes reporta la obtención de una mayor fiabilidad del sistema, junto con unas dimensiones más reducidas del mismo y, además, permite al diseñador concentrar su esfuerzo en la concepción lógica del sistema.

5.4.2 Circuitos digitales con componentes discretos: lógica resistencia-transistor

El primer circuito que se utilizó fue el de la figura 5.11 que representa una puerta NO-O en lógica negativa realizada con resistencias y un transistor bipolar de germanio. El principal inconveniente de este circuito es la necesidad de utilizar una alimentación auxiliar para polarizar inversamente la base del transistor cuando

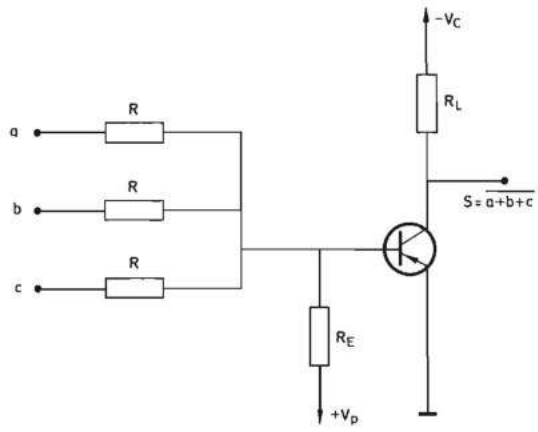


FIGURA 5.11.—**Puerta NO-O en lógica negativa realizada con resistencias y un transistor PNP de germanio.**

todas las entradas del mismo se encuentran en estado cero a fin de reducir la corriente de pérdidas del colector. Este inconveniente se evitó posteriormente mediante la utilización de transistores de silicio, tal como se indica en la figura 5.12; la resistencia R_E se conecta al cero de la tensión $+V_p$ y, por tanto, se suprime la tensión de polarización $-V_c$. Otra desventaja del circuito es que la corriente

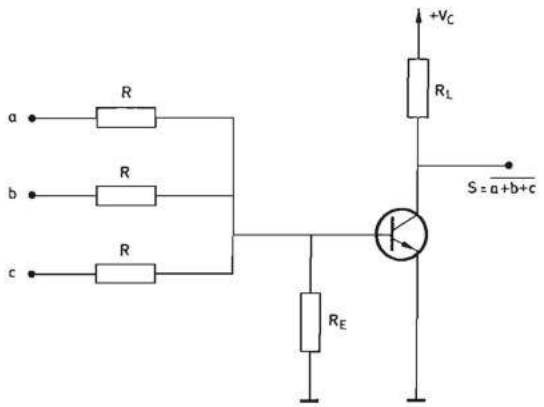


FIGURA 5.12.—**Puerta NO-O en lógica positiva realizada con resistencias y un transistor NPN de silicio.**

de base proporcionada por una sola entrada debe de ser suficiente para saturar el transistor, por lo cual, al poner más de una entrada en estado uno se produce una sobresaturación que disminuye la frecuencia máxima de conmutación del circuito.

Una ventaja importante es la gran inmunidad al ruido.

5.4.3 Circuitos bloque: tecnología resistencia-transistor (RTL)

Constituyeron un paso adelante en la concepción de los circuitos digitales electrónicos.

Consistieron en encapsular juntos los componentes de la puerta NO-O (NOR) de la figura 5.12, constituyendo de esta forma un bloque que el diseñador de sistemas digitales podía utilizar como tal.

En esta tecnología se realizaron además circuitos temporizadores cuyo diagrama de bloques se representa en la figura 5.13*b* junto con la secuencia de evolución en el tiempo de sus señales. Como se puede observar, la principal característica de este circuito es que la salida cambia inmediatamente del estado 1 al 0 cuando la entrada conmuta del 0 al 1 y, por el contrario, cuando conmuta del estado 1 al 0, la salida pasa del estado 0 al 1 después de un cierto tiempo $t = RC$ variable entre amplios márgenes modificando una resistencia y un condensador que se conectan exteriormente. Estos circuitos se aplicaron extensamente en la realización de sistemas de control industrial.

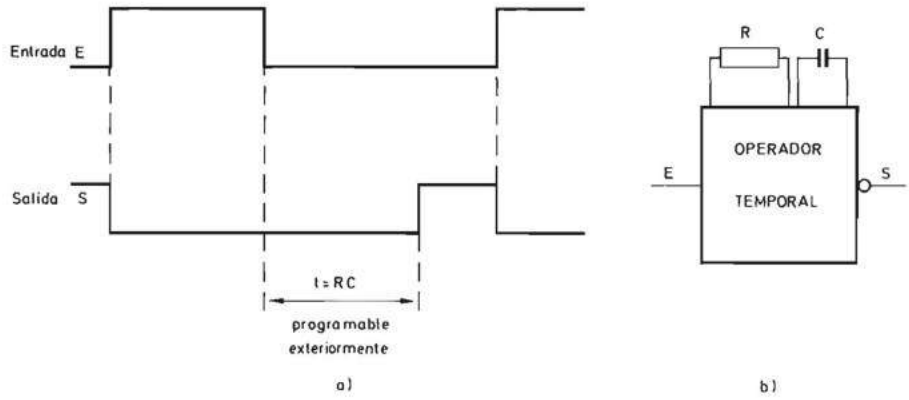


FIGURA 5.13.—**Circuito temporizador: *a*) diagrama de evolución en el tiempo de sus señales; *b*) símbolo lógico.**

La frecuencia máxima de funcionamiento de los circuitos bloque de tecnología resistencia-transistor quedó limitada a 50 kHz aproximadamente y por ello en la práctica su utilización se limitó a la realización de sistemas secuenciales de control asíncronos.

5.4.4 Circuitos integrados digitales

5.4.4.1 Introducción. El progreso de la tecnología de fabricación de los semiconductores permitió, a principios de la década de 1960, la difusión de varios componentes en un único sustrato semiconductor. Apareció así el «circuito integrado monolítico» (en inglés «Monolithic Integrated Circuit») y se inició el desarrollo de la microelectrónica que es el área de la electrónica aplicada dedicada al estudio de los métodos y procesos de realización de circuitos integrados.

La existencia en la electrónica digital de numerosos bloques funcionales la hizo idónea para la realización de circuitos integrados monolíticos de complejidad creciente y contribuyó al espectacular desarrollo de la microelectrónica a partir de finales de la década de 1960.

En sucesivos apartados se estudian los principales conceptos de los circuitos integrados digitales. Un análisis más profundo de la evolución y del estado actual de los circuitos integrados se sale fuera de los límites de este libro y por ello se remite al lector a la bibliografía [GEIG 90] [HURS 85].

5.4.4.2 Clasificación de los circuitos integrados digitales. La gran cantidad de conceptos referentes a las características de los dispositivos, de los circuitos y de los sistemas electrónicos digitales ha dado lugar a numerosos tipos de circuitos integrados digitales.

El primer concepto utilizable para clasificarlos es la forma en que se realiza el circuito integrado que puede ser monolítico o híbrido. Un circuito integrado mono-

lítico está constituido por un conjunto de dispositivos colocados en un único soporte o sustrato semiconductor que constituye uno de los elementos del mismo. Un circuito integrado híbrido está formado por un conjunto de circuitos integrados monolíticos y componentes discretos que se colocan sobre un determinado soporte inerte que sirve exclusivamente para realizar las conexiones entre ellos.

En sucesivos apartados se analizan las características generales de los circuitos integrados monolíticos e híbridos.

A continuación se analizan las distintas formas de realizar los circuitos integrados monolíticos que han evolucionado vertiginosamente para dar lugar al extraordinario desarrollo de la electrónica digital.

5.4.4.2.1 Circuitos integrados digitales monolíticos. Los circuitos integrados digitales monolíticos (en adelante CIDM) se pueden clasificar de acuerdo con dos grandes conceptos interrelacionados, tal como se indica en la figura 5.14:

a) Según la forma en que se realizan físicamente, que a su vez da lugar a:

— Según el tipo de dispositivos utilizado, los CIDM bipolares, MOS y BICMOS.
— Según el número de dispositivos colocados en el interior de un único circuito, los CIDM SSI, MSI, LSI, VLSI, ULSI y GLSI.

b) Según la forma en que se realiza el diseño desde el punto de vista del ingeniero de aplicaciones:

— Los CIDM normalizados o estándar (standard off-the-shelf integrated circuits).
— Los CIDM especificados por el usuario.

En sucesivos apartados se estudian los diferentes tipos de CIDM, pero la gran complejidad de los CIDM hace que su presentación secuencial en un libro entrañe muchas dificultades. Por ello, en este libro se incluye un CD-ROM que contiene un sistema hipermedia que presenta los CIDM de forma interactiva para que el lector pueda elegir la secuencia de análisis que se adapte mejor a sus conocimientos y acceder en cada instante de forma sencilla a cualquiera de los diversos conceptos interrelacionados relativos a los CIDM.

5.4.4.2.1.1 Clasificación según el tipo de dispositivo utilizado. Si se utilizan transistores bipolares se obtienen los CIDM bipolares que fueron los primeros en desarrollarse, y han tenido una gran evolución con la finalidad de reducir al máximo el producto del tiempo de propagación por la potencia disipada. En la figura 5.14 se indican por sus siglas las principales familias de circuitos integrados digitales bipolares que se estudian en el apartado 5.4.4.3.

El otro dispositivo que se utiliza para realizar los CIDM es el transistor de efecto de campo de puerta aislada [Metal Oxide Semiconductor (MOS)]. Por sus

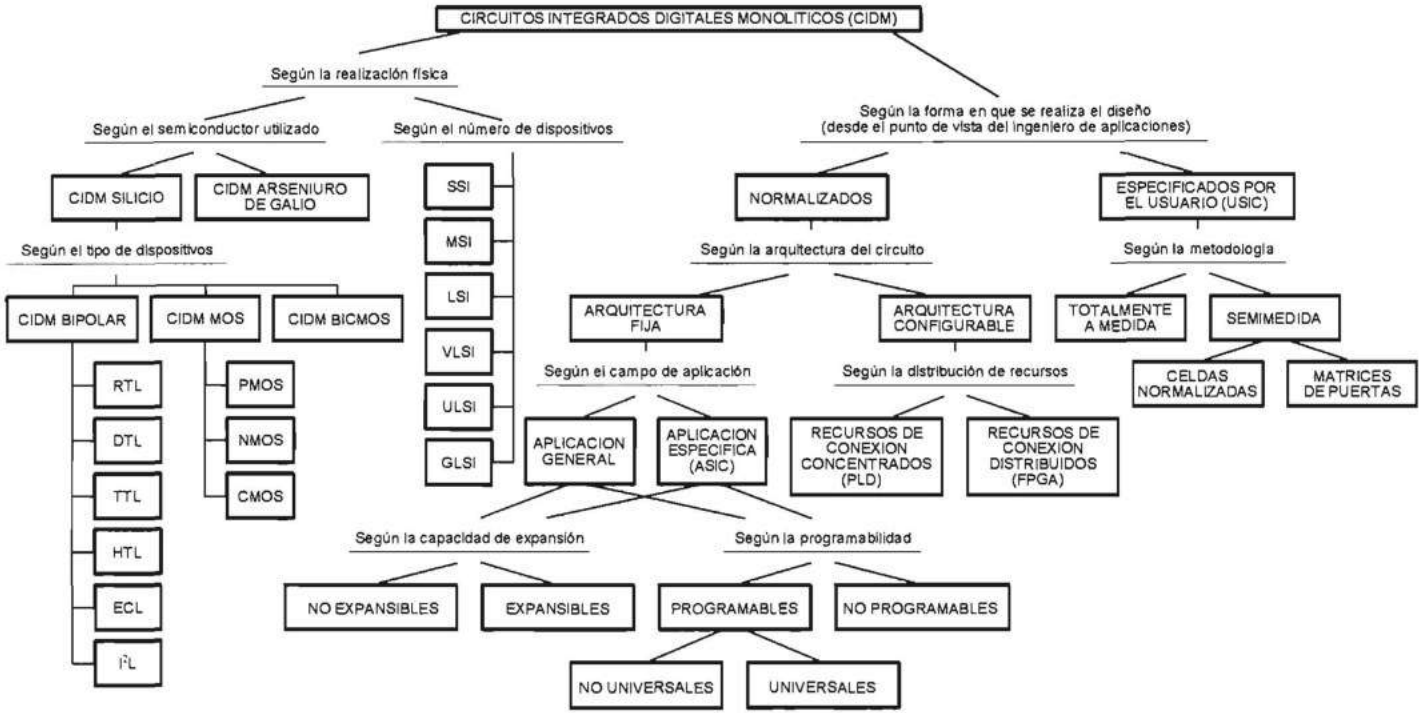


FIGURA 5.14.—**Clasificación de los CIDM.**

características los transistores MOS han permitido la integración de gran número de componentes en un único circuito integrado monolítico. Al estudio de las tecnologías MOS se dedica el apartado 5.4.4.4.

La combinación de transistores MOS y bipolares permite realizar circuitos con gran cargabilidad de salida (fan-out) y a su estudio se dedica el apartado 5.4.4.5.

El progreso de la física del estado sólido ha permitido el uso de compuestos III-V, y en concreto del arseniuro de galio, para realizar circuitos integrados digitales que se estudian en el apartado 5.4.4.6.

5.4.4.2.1.2 Clasificación según el número de dispositivos. Tanto los CIDM de silicio como los de arseniuro de galio se pueden a su vez clasificar según el número de dispositivos o de puertas básicas contenidas en los mismos.

El progreso de las técnicas de integración y la aplicación de métodos de diseño asistido por computador han hecho que el número de componentes que se pueden difundir en un solo sustrato semiconductor haya ido aumentando paulatinamente (tabla 5.1).

Nivel de integración	*N.º de componentes*	*N.º de puertas*
Pequeña escala de integración (SSI)	10 a 100	1 a 10
Mediana escala de integración (MSI)	100 a 1.000	10 a 100
Gran escala de integración (LSI)	1.000 a 10.000	100 a 1.000
Muy gran escala de integración (VLSI)	10.000 a 100.000	1.000 a 10.000
Ultra gran escala de integración (ULSI)	100.000 a 1.000.000	10.000 a 10.000
Giga gran escala de integración (GLSI)	> 1.000.000	> 100.000

TABLA 5.1.—**Clasificación de los circuitos digitales monolíticos de acuerdo con el número de componentes que contienen.**

Se comenzó integrando puertas elementales NO-Y y NO-O en un número de una a seis en función del número de entradas, y biestables del tipo *J-K* principal subordinado (Master-Slave), *D* activado por flancos (edge-triggered) y *D* cerrojo (latch) (ver capítulo 6). Estos circuitos constituyen la pequeña escala de integración (SSI), denominación que engloba a los circuitos integrados que contienen entre 10 y 100 componentes, equivalentes aproximadamente a 1 y 10 puertas lógicas respectivamente.

La existencia de muchos sistemas digitales complejos normalizados tales como contadores, multiplexores, decodificadores, etc., ha permitido su diseño en circuito integrado y produjo la aparición de la escala de integración media (MSI). Bajo esta denominación se incluyen aquellos circuitos integrados que contienen de 100 a 1000 componentes, equivalentes aproximadamente a 10 y 100 puertas lógicas respectivamente.

El desarrollo combinado de la teoría de los bloques funcionales y de las técnicas microelectrónicas condujo a los circuitos de gran escala de integración (LSI), que

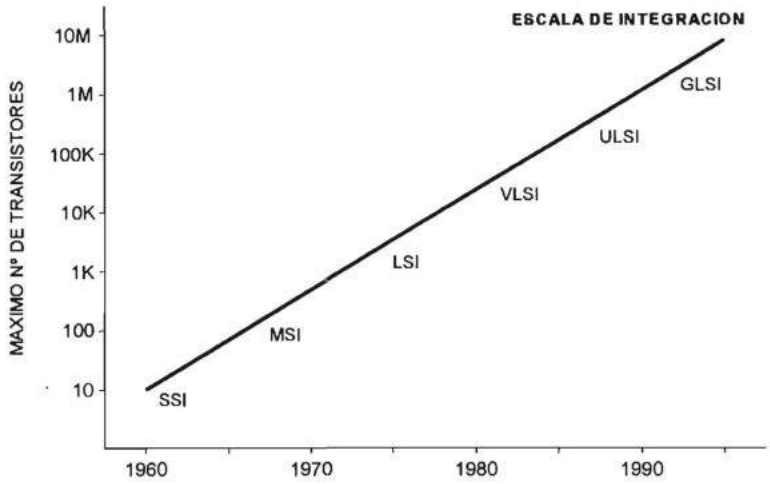


FIGURA 5.15.—**Representación gráfica de la evolución de la capacidad de integración.**

poseen un número de puertas comprendido entre 100 y 1000, y dio lugar al nacimiento del microprocesador.

El progreso continuo de las técnicas de integración permitió la realización de circuitos de muy gran escala de integración (VLSI), de ultra gran escala de integración (ULSI) y de giga gran escala de integración (GLSI) que llegan a contener más de 1.000.000 de componentes (equivalentes aproximadamente a 100.000 puertas) en su interior.

El vertiginoso desarrollo de los circuitos integrados se pone en evidencia mediante el gráfico semilogarítmico de la figura 5.15 que muestra que desde 1960

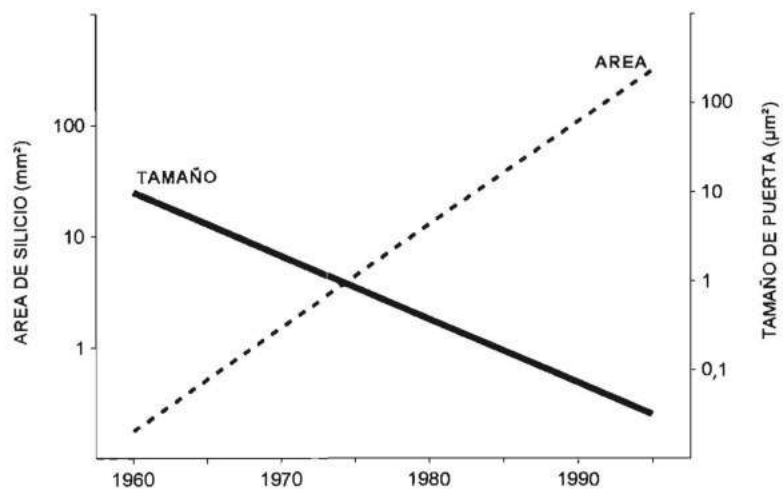


FIGURA 5.16.—**Representación gráfica de la evolución del área de los CIDM y el tamaño de las puertas lógicas.**

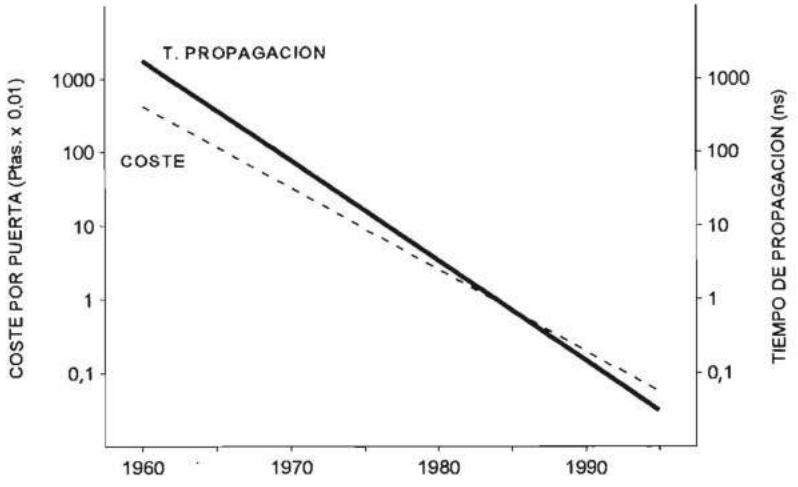


FIGURA 5.17.—**Representación gráfica de la evolución del tiempo de propagación y el coste por puerta de los CIDM.**

hasta 1990 se ha multiplicado aproximadamente por 100.000 el número de transistores que los fabricantes han sido capaces de colocar en un único circuito integrado monolítico. Este desarrollo ha sido consecuencia de la disminución del tamaño de la puerta lógica y del aumento del área del circuito integrado, que se indican en la figura 5.16. Otra consecuencia de ello es la disminución del tiempo de propagación de una puerta básica que se ha dividido por mil y del coste de la misma que es actualmente próximo a los diez céntimos tal como se indica en la figura 5.17.

El resultado de todo ello es que en la década de los años 1990 se pueden realizar sistemas digitales que en la década de los años 1960 ni siquiera se podían imaginar.

5.4.4.2.1.3 Clasificación de los circuitos normalizados o estándar. Reciben el nombre de normalizados, estándar o comerciales (en inglés «Standard off-the-shelf integrated circuits») aquellos CIDM que son suministrados por los fabricantes con unas características eléctricas y funcionales perfectamente definidas y tienen asignadas unas siglas formadas por un conjunto de números y/o letras. Un ejemplo típico es la familia 74 de circuitos integrados TTL.

Según la forma (arquitectura) en que está realizado el sistema físico (hardware) que los constituye, los CIDM estándar se clasifican en dos grandes clases:

a) CIDM de arquitectura fija, que son aquellos que, como su nombre indica, poseen un sistema físico (hardware) en el que todos sus elementos están unidos rígidamente entre sí y funcionan en algún instante a lo largo de su vida útil.
b) CIDM de arquitectura configurable, que son aquellos cuyo sistema físico (hardware) funcional se puede variar modificando el estado de unas variables binarias.

Ambos tipos de CIDM se estudian a continuación.

CIDM de arquitectura fija

Los CIDM de arquitectura fija pueden a su vez ser de aplicación general o de aplicación específica. Los CIDM de aplicación general son aquellos que constituyen bloques funcionales que se utilizan en la realización de numerosos sistemas digitales. Existen circuitos de este tipo tanto en pequeña escala de integración (SSI) [por ejemplo el circuito 7400 de tecnología TTL que contiene 4 puertas NO-Y (NAND) de cuatro entradas], en mediana escala de integración (MSI) (por ejemplo el circuito 74LS194 que es un registro de desplazamiento cuyo símbolo normalizado se representa en la figura A1.60 del apéndice 1), en gran escala de integración (LSI) (por ejemplo los microprocesadores de 8 bits como el 6502 de Rockwell, el 6800 de Motorola o el 8085 de Intel), en muy gran escala de integración (VLSI) (por ejemplo los microprocesadores de 16 bits 8086 de Intel y 68000 de Motorola), en ultra gran escala de integración (ULSI) (por ejemplo el microprocesador 486 de Intel) y en giga gran escala de integración (GLSI) (por ejemplo el microprocesador Pentium de Intel).

Pero el avance de la microelectrónica hizo que fuese posible introducir en un único CIDM un sistema digital completo que realiza una función compleja determinada o específica, o una parte de un sistema digital complejo. Son ejemplos del primer caso un sistema de riego automático o un sistema de telemando, y del segundo caso las unidades de interfaz complejas de procesadores digitales programables (como una unidad de comunicaciones, una unidad de visualización gráfica, etc.). Esto hizo que los fabricantes de circuitos integrados comercializasen, como normalizados, circuitos de este tipo que realizan una función específica.

No existe una denominación de estos circuitos universalmente aceptada. Algunos fabricantes los denominan circuitos integrados de aplicación específica normalizados [en inglés «Standard Application Specific Integrated Circuits» (Standard ASIC)] y otros los denominan productos normalizados de aplicación específica [en inglés «Application Specific Standard Products» (ASSP)].

Tanto los CIDM de arquitectura fija de aplicación general como los de aplicación específica se pueden clasificar de acuerdo con dos conceptos interrelacionados:

a) La expansibilidad.

Los CIDM expansibles son aquellos que tiene una determinada capacidad (por ejemplo una unidad operativa de 8 bits) pero que poseen los recursos lógicos y terminales adecuados para acoplar varios entre sí (sin necesidad de añadir otros circuitos) y obtener de esta forma un conjunto de mayor capacidad. Fueron el resultado del progreso de la microelectrónica y se desarrollaron a nivel LSI y VLSI. Recibieron por ello el nombre de expansibles (en inglés «Bit slice»). Un ejemplo característico es la unidad operativa de 4 bits Am 2903 de AMD que se estudia en el segundo tomo de este libro dedicado a los procesadores digitales. El interés de los CIDM expansibles disminuyó al alcanzarse la ULSI y ser posible realizar en un único CIDM un procesador de 32 o 64 bits.

b) La programabilidad.

Los CIDM no programables realizan una única función con unas características perfectamente definidas. Un ejemplo es el circuito 7400 citado anteriormente.

Por el contrario, los circuitos digitales programables son aquellos cuya función se puede variar modificando el valor de unas variables binarias. Los CIDM programables pueden, a su vez, ser o no universales.

Los CIDM de arquitectura fija programables no universales realizan más de una función o una función compleja que admite varias variantes y constituyen bloques funcionales de aplicación general. Estos CIDM se caracterizan por poseer un conjunto de variables binarias de entrada que suelen denominarse variables de modo de operación o de control y permiten seleccionar las funciones o variantes indicadas anteriormente. Son ejemplos de CIDM de función fija programables no universales la unidad aritmética y lógica 74181 (estudiada en el capítulo 5) y el circuito 74LS194 (ver figura A1.60 del apéndice 1) que es un registro de desplazamiento que posee cuatro modos de operación seleccionables mediante dos variables binarias.

Los CIDM de función fija programables universales contienen la unidad central de proceso (CPU) de un computador (como por ejemplo el 8085, 8086, 80286, 80386, 80486 o el Pentium de Intel, el 6800 o el 68000 de Motorola, etc.), la totalidad de un computador (por ejemplo 8051 o el 80196 de Intel o el 68HC11 de Motorola) o un procesador digital de señales (por ejemplo el DSP56002 de Motorola). Todos ellos se caracterizan porque la función que realizan se modifica cambiando el contenido de una memoria de acceso aleatorio.

Los circuitos integrados digitales estándar presentan, independientemente de su escala de integración, las siguientes características:

— Reducido coste gracias a la fabricación en grandes series.
— Gran fiabilidad debido también a la fabricación en grandes series.
— Disponibilidad de varias fuentes de suministro.
— Escasa protección contra copias no autorizadas, excepto los programables universales que contienen en su interior la totalidad de un computador.

El diseño de sistemas digitales con CIDM estándar de arquitectura fija se caracteriza por la realización de un prototipo con los circuitos adecuados elegidos por el diseñador y la verificación de su comportamiento mediante la utilización de los instrumentos electrónicos adecuados (osciloscopios, analizadores lógicos, emuladores, etc.).

CIDM configurables

Los CIDM de arquitectura fija, tanto si son expansibles como si no lo son e independientemente de su nivel de integración, poseen un conjunto de recursos lógicos conectados rígidamente entre sí que se utilizan en su totalidad a lo largo de la vida útil de los mismos. La programabilidad les da a estos CIDM la característica de poder cambiar su forma de operación modificando el contenido de una memoria

de acceso aleatorio o las conexiones de una matriz lógica programable. Pero la flexibilidad así obtenida no es suficiente en algunas aplicaciones como por ejemplo los circuitos de interfaz de ciertos periféricos de un procesador. Por ello surgió el interés por desarrollar unos CIDM cuyo sistema físico (hardware) funcional pudiera ser modificado mediante el estado de unas variables binarias.

La configurabilidad es un concepto asociado a los sistemas electrónicos digitales cuya función se puede modificar utilizando solamente una parte de los elementos que los componen y/o cambiando la interconexión entre ellos. Dicha modificación se lleva a cabo mediante la programación del estado de un conjunto de variables binarias independientes o asociadas entre sí formando una determinada estructura de memoria.

En la figura 5.18*a* se representa gráficamente un ejemplo de circuito digital configurable en el que los bloques lógicos están formados por elementos lógicos (en este caso puertas Y) enlazados entre sí a través de dispositivos electrónicos cuyo estado de conducción o no conducción se controla mediante una variable binaria. Los bloques lógicos se conectan además entre sí a traves de recursos de interconexión que contienen también dispositivos electrónicos como los que se acaban de describir, cuya finalidad es controlar las conexiones entre los mismos.

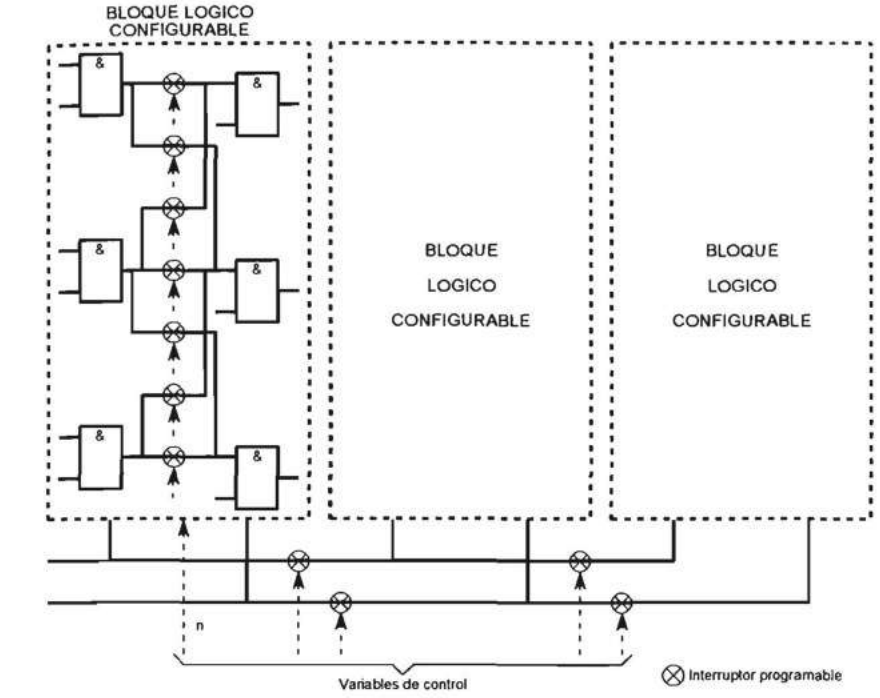
De lo expuesto se deduce que los circuitos digitales configurables contienen un elevado número de elementos lógicos en su interior, de los cuales sólo se utiliza en cada aplicación concreta un porcentaje que en la mayoría de los casos no supera el 60% y por ello sólo han sido posibles al alcanzarse la muy gran escala de integración (VLSI) y la ultra gran escala de integración (ULSI).

El interés de la utilización de circuitos digitales configurables por parte de cualquier diseñador que tenga que realizar sistemas electrónicos digitales complejos ha hecho que diversos fabricantes de circuitos integrados los ofrezcan comercialmente como circuitos normalizados (en inglés «Standard off-the-shelf integrated circuits»). La configuración del circuito se realiza a través de determinados terminales externos que, por medio de los adecuados recursos lógicos, actúan sobre los dispositivos electrónicos programables situados en los bloques lógicos configurables o en los recursos de interconexión (figura 5.18*b*) y los hacen conducir o los ponen en estado de corte.

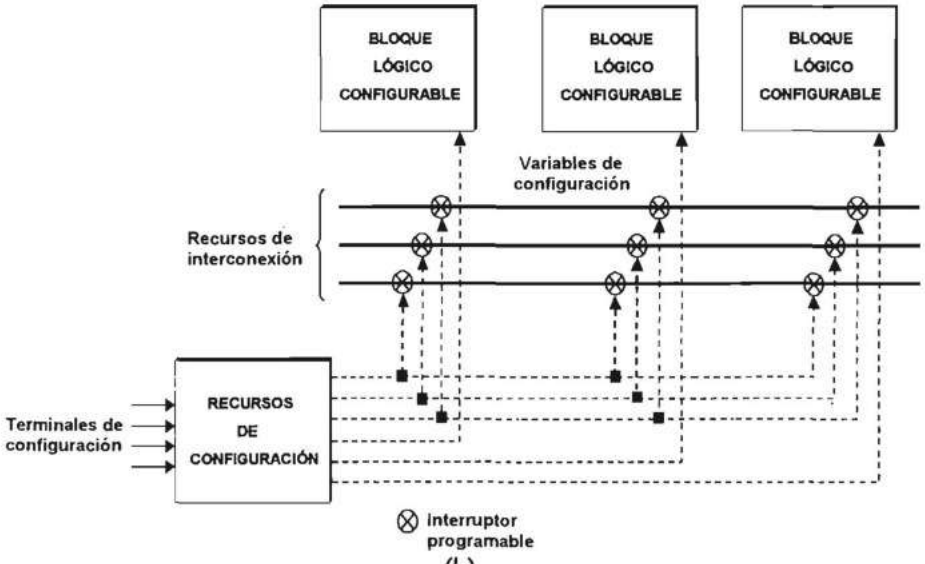
Para realizar un CIDM configurable es necesario:

— Distribuir espacialmente los recursos lógicos divididos en bloques, que facilitan la realización de circuitos digitales de elevada complejidad.
— Distribuir espacialmente los recursos de interconexión entre los bloques lógicos para facilitar el enlace entre los mismos.

La mayor o menor concentración de recursos lógicos se denomina granularidad. La granularidad de un circuito configurable es elevada si posee un gran número de bloques lógicos de reducida complejidad cada uno, y es baja en caso contrario. Una elevada granularidad trae consigo la necesidad de disponer de complejos recursos de interconexión pero proporciona una gran flexibilidad que no es posible con una granularidad reducida. Por ello, existe una gran cantidad de formas diferentes de distribuir los recursos lógicos y los de interconexión, que trae como consecuencia la existencia de múltiples soluciones por parte de diferentes fabricantes.



(a)



(b)

FIGURA 5.18.—**Fundamentos de los CIDM configurables: *a*) ejemplo de arquitectura; *b*) esquema global con recursos de configuración.**

La gran cantidad de circuitos digitales configurables diferentes es debida no solamente a la característica antes citada, sino también a diversas razones comerciales como la imagen de marca, la búsqueda de mercado cautivo, etc.

Por otra parte, la necesidad de utilizar recursos de diseño asistido por computador para configurar los circuitos, aumenta el interés de los fabricantes por poseer arquitecturas propias y les permite no proporcionar una información exhaustiva sobre la realización física de los mismos.

El desarrollo de los circuitos digitales configurables se realizó siguiendo dos grandes tendencias:

— A partir de las matrices lógicas programables PLA y PAL (ver capítulo 3), añadiéndoles recursos lógicos cuya función fuese modificable mediante la programación del estado de unas variables binarias.
— A partir de las estructuras utilizadas en la síntesis de circuitos integrados digitales semimedida (ver apartado 5.4.4.2.1.4 más adelante), haciendo configurables los bloques lógicos y las interconexiones entre ellos, mediante la selección del estado de alta o baja impedancia de determinados dispositivos electrónicos.

Estas dos tendencias permiten clasificar los circuitos digitales configurables en dos grandes organizaciones, indicadas en la tabla 5.2, que se caracterizan por la forma en que se implementan los recursos de interconexión:

a) Organización de los circuitos digitales configurables con recursos de interconexión concentrados.

Esta organización, cuyo diagrama de bloques general se representa en la figura 5.19, tiene sus antecedentes en las matrices lógicas programables y por ello suele

CIRCUITOS DIGITALES CONFIGURABLES	Con recursos de interconexión concentrados (PLD)	DLP básicos (BPLD)
		DLP avanzados (APLD)
	Con recursos de interconexión semiconcentrados o semidistribuidos	DLP complejos (CPLD)
		Cuadrícula
	Con recursos de interconexión distribuidos (FPGA)	Terraza
		Mar de puertas

TABLA 5.2.—**Organización de los circuitos digitales configurables.**

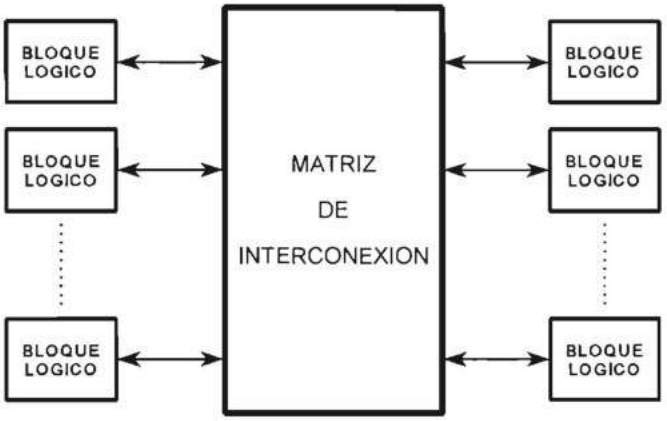


FIGURA 5.19.—**Organización matricial.**

recibir el nombre de matricial. Los circuitos incluidos en ella se denominan dispositivos lógicos programables (DLP) [en inglés «Programmable Logic Devices» (PLD)] y han tenido un gran desarrollo desde que, a mediados de la década de 1970, se realizaron los primeros secuenciadores lógicos programables (en inglés «Programmable Logic Sequencers» (PLS), también denominados por algunos fabricantes matrices lógicas con registros (en inglés «Registered PAL»). Por ello su estudio sistemático tiene también un gran interés práctico y a él se dedicaron diversos trabajos en el departamento de Tecnología Electrónica de la Universidad de Vigo que los clasifican en básicos, avanzados y complejos [ALVA 95] [PERE 91].

Los DLP básicos (en inglés BPLD), cuyo diagrama de bloques general se representa la figura 5.20, están formados por una matriz PAL realimentada a través de unos bloques lógicos denominados macroceldas [ALTE 88] [INTE 90] [PHIL 90]. Las limitaciones de los DLP básicos hicieron que diversos fabricantes desarrollasen los DLP avanzados (en inglés APLD) siguiendo dos grandes líneas:

— Dotando a los DLP básicos de recursos lógicos destinados a que las macroceldas compartan recursos de la matriz PAL (figura 5.21) [AMD 89] [ALTE 90].
— Dotando a los DLP básicos de varias matrices de interconexión (figura 5.22) [INTE 90].

Finalmente, los DLP complejos (en inglés CPLD) [BURS 96] [LATT 94] [LATT 95] son el resultado de distribuir los recursos de interconexión para aumentar la flexibilidad, y se les puede considerar como una clase de circuitos digitales configurables que poseen recursos de interconexión semiconcentrados o semidistribuidos (tabla 5.2).

La elevada complejidad de los DLP complejos hace que existan múltiples formas de realizarlos, una de las cuales se indica en la figura 5.23, que representa un DLP complejo formado por un conjunto de bloques lógicos interconectados no sólo

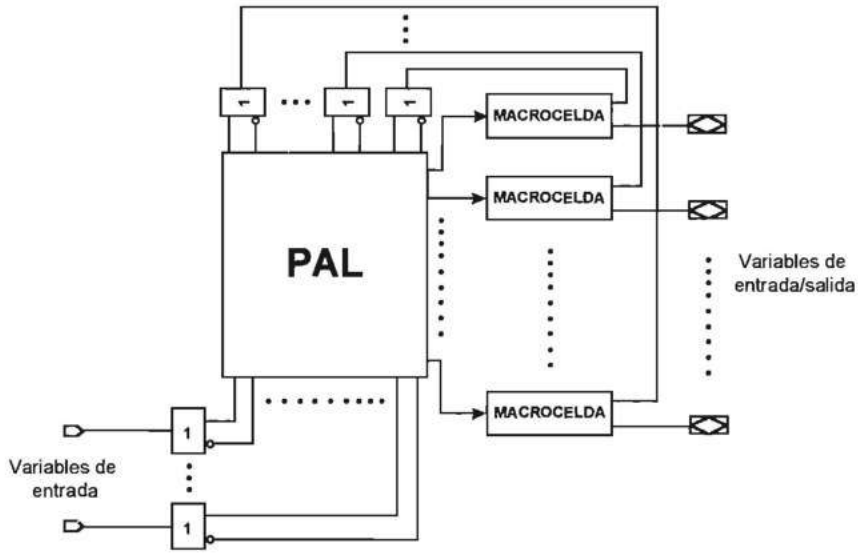


FIGURA 5.20.—**Diagrama de bloques general de un DLP básico.**

a través de una matriz global sino también a través de matrices locales que los conectan además con los terminales de entrada/salida. Algunos fabricantes de DLP complejos los denominan conjuntos configurables de puertas [en inglés «Field Programmable Gate Arrays» (FPGA) [INTE 94]. Por ello en la tabla 5.2 se considera que forman parte simultáneamente de ambos tipos de organizaciones.

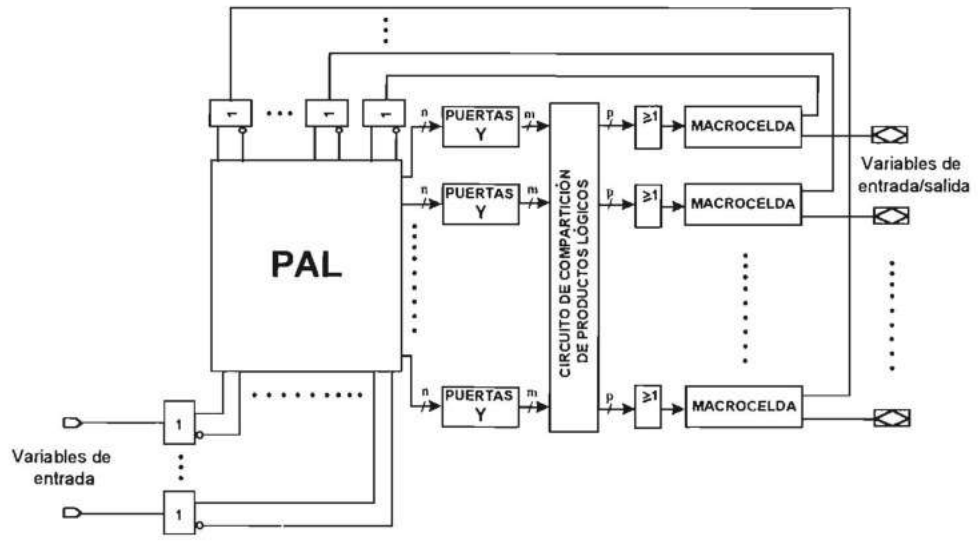


FIGURA 5.21.—**Diagrama de bloques de un DLP avanzado con compartición de productos lógicos.**

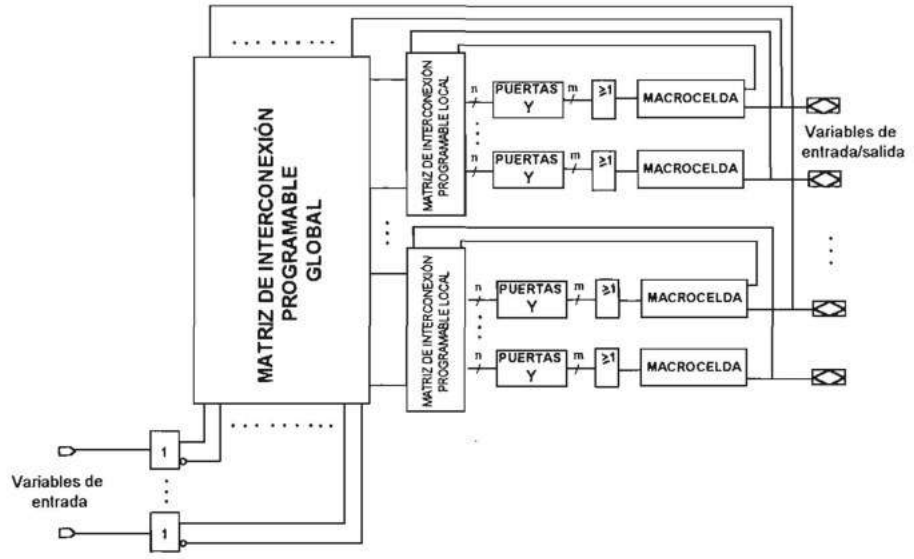


FIGURA 5.22.—**Diagrama de bloques de un DLP avanzado (ADLP) con varias matrices de interconexión.**

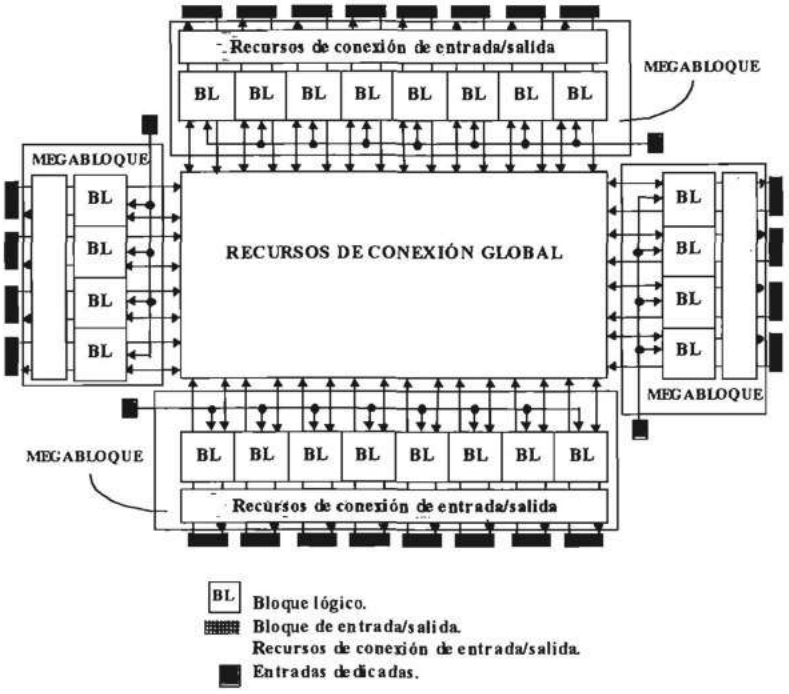


FIGURA 5.23.—**Diagrama de bloques de un DLP complejo (CPLD).**

b) Organización con recursos de interconexión distribuidos.

Esta organización tiene sus antecedentes en los circuitos integrados semimedida que se estudian en el apartado siguiente. Los circuitos que la utilizan suelen recibir la denominación de conjuntos configurables de puertas [en inglés «Field Programmable Gate Arrays (FPGA)].

El vasto campo de aplicación de los circuitos digitales configurables con recursos de interconexión distribuidos ha hecho que sean numerosos los grupos de investigadores que se han dedicado a utilizarlos así como los fabricantes que los han implementado a partir de la mitad de la década de 1980 [ACTE 95] [AT&T 95] [XILI 96].

Los circuitos digitales configurables con recursos de interconexión distribuidos están formados por cuatro elementos básicos: los bloques lógicos internos, los recursos de comunicación, los bloques lógicos de entrada y salida y los recursos de interconexión.

Según la forma en que se organizan los bloques lógicos internos se distinguen tres grandes organizaciones de FPGA:

FPGA de organización tipo terraza

Esta organización se caracteriza por tener los bloques lógicos internos dispuestos en filas separadas por canales en los que se ubican los recursos de interconexión horizontal, tal como se muestra en la figura 5.24 [CROS 92] [TEXA 93]. Por ello recibe el nombre de terraza (en inglés «Terraced or Row based organization»). Los bloques lógicos internos pueden tener una menor o mayor complejidad, lo cual da lugar a estructuras de granularidad fina o gruesa respectivamente. La conexión en-

FIGURA 5.24.—**Circuitos digitales configurables de organización tipo terraza o de filas.**

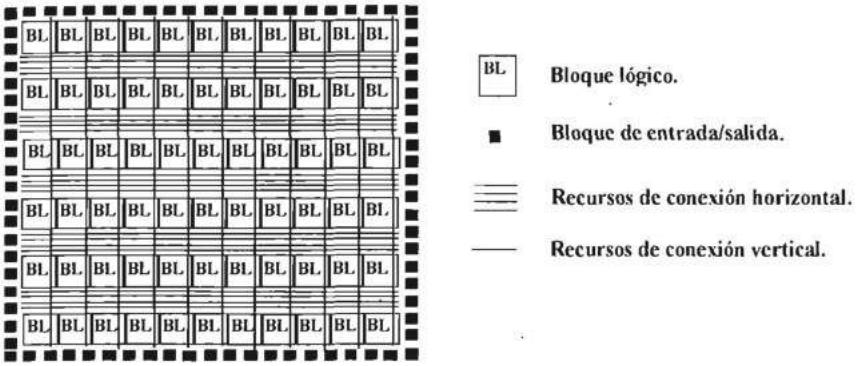


FIGURA 5.25.—**Organización tipo terraza.**

tre estos bloques se realiza mediante recursos de interconexión horizontal y vertical. Las señales de entrada y salida de los bloques lógicos se conectan a través de recursos de interconexión horizontal que existen en mayor número que los de interconexión vertical. Por su parte, los de interconexión vertical facilitan la comunicación entre los canales horizontales y transmiten señales globales como, por ejemplo, la de reloj. En la figura 5.25 se representa el diagrama de bloques de una FPGA de organización tipo terraza.

FPGA de organización tipo cuadrícula

Esta organización está constituida por un conjunto de bloques lógicos internos dispuestos en forma de filas y columnas delimitadas por los recursos de interconexión que se extienden de forma vertical y horizontal entre los bloques, tal como se muestran en la figura 5.26. Al igual que en el caso anterior, la complejidad de cada

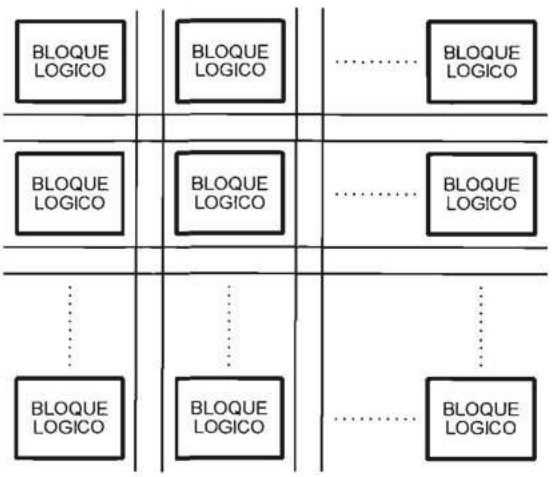


FIGURA 5.26.—**Circuitos digitales configurables de organización tipo cuadrícula.**

bloque lógico puede variar considerablemente entre los diferentes fabricantes [ALTE 95] [XILI 94].

FPGA de organización tipo mar de puertas

Esta organización presenta, con respecto a la cuadrícula, la similitud de que los recursos lógicos se organizan en filas y columnas (figura 5.27*a*). Por el contrario, se diferencian de ella en que los recursos de interconexión se sitúan en capas diferentes de las de los bloques lógicos, tal como se muestra en la figura 5.27*b* [XILI 95]. En una primera capa se localizan los bloques lógicos configurables que se denominan células lógicas por ser, en general, elementos lógicos muy simples que permiten implementar puertas lógicas y funciones sencillas. La ausencia de recursos y de interconexión en esta capa permite un mejor aprovechamiento del área, por lo que las FPGA con esta estructura presentan una gran densidad de recursos lógicos y una granularidad fina.

En la segunda y tercera capa se ubican los recursos de interconexión verticales y horizontales respectivamente. El elevado número de recursos de conexión disponibles en esta estructura permite que las salidas de una célula puedan conectarse a las entradas de cualquier otra célula. De esta forma se obtiene un aprovechamiento de recursos cercano al 100%, lo que hace que esta estructura se asemeje a los conjuntos de puertas programables por máscara descritos en el apartado siguiente. Además, las facilidades de conexionado de esta organización permiten disminuir apreciablemente los retardos, por lo que su velocidad de operación es menos sensible a las decisiones de partición (en inglés «mapping») y de ubicación (en inglés «placement»).

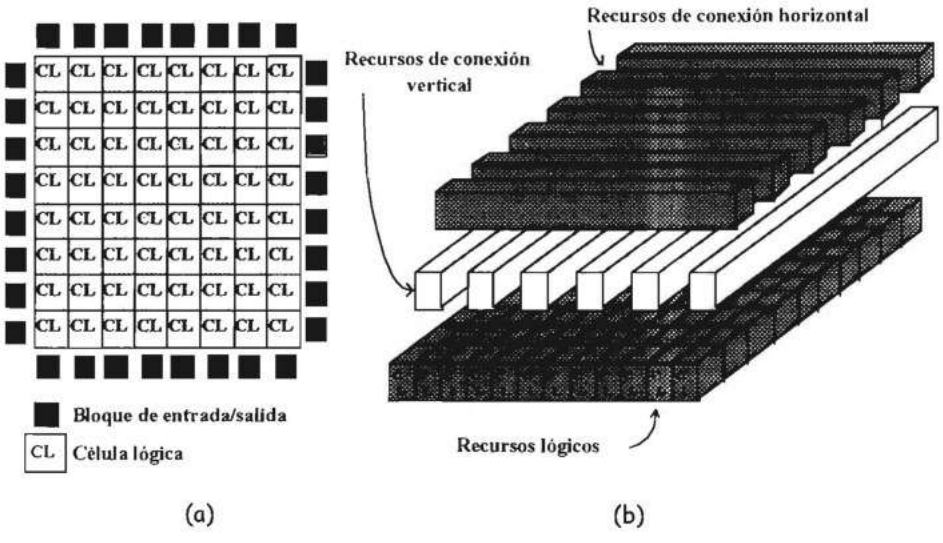


FIGURA 5.27.—**Organización tipo mar de puertas: *a*) estructura de puertas; *b*) estructura en capas.**

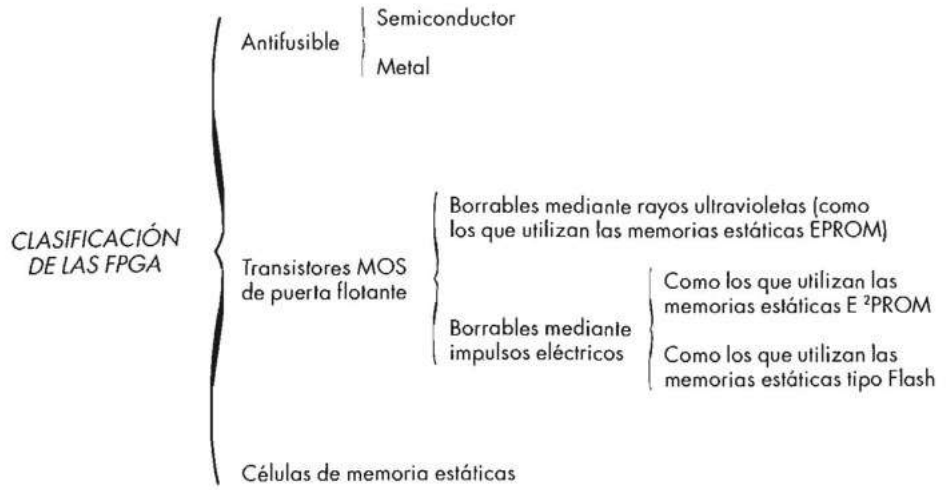


TABLA 5.3.—**Clasificación de las FPGA de acuerdo con la tecnología de los interruptores configurables.**

Las FPGA se pueden además clasificar de acuerdo con la tecnología de los elementos físicos configurables, tal como se indica en la tabla 5.3.

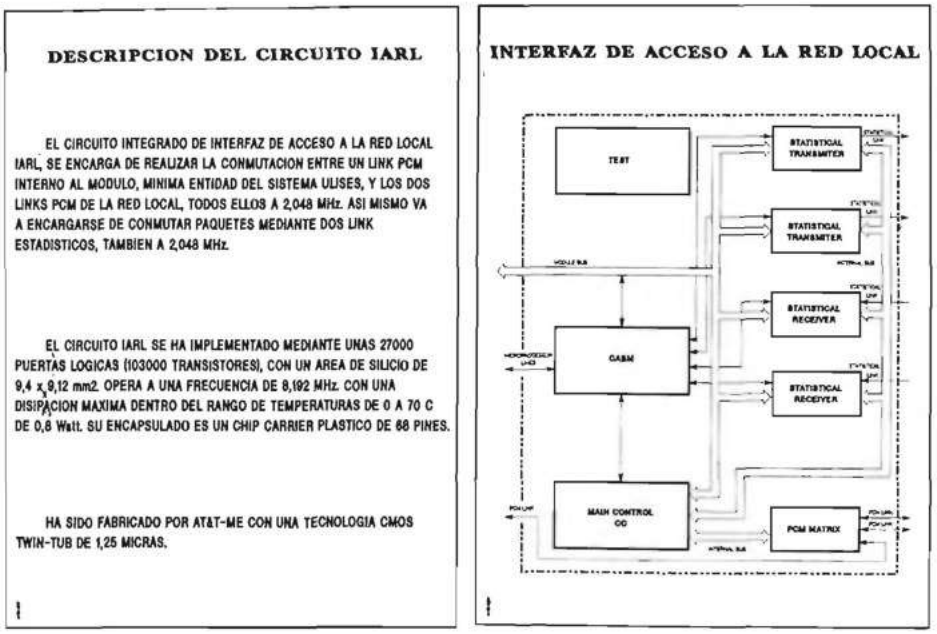
Las FPGA que utilizan antifusibles sólo se pueden configurar una vez, mientras que las basadas en transistores MOS de puerta flotante o en células de memoria activa estática (ver capítulo 7) se pueden configurar múltiples veces.

Un estudio más detallado de los CIDM configurables se sale fuera de los límites de este libro. Por ello, al lector interesado se le remite a los libros «Circuitos digitales configurables» tomos I y II publicados por varios profesores del departamento de Tecnología Electrónica de la Universidad de Vigo y miembros del Instituto de Electrónica Aplicada Pedro Barrié de la Maza.

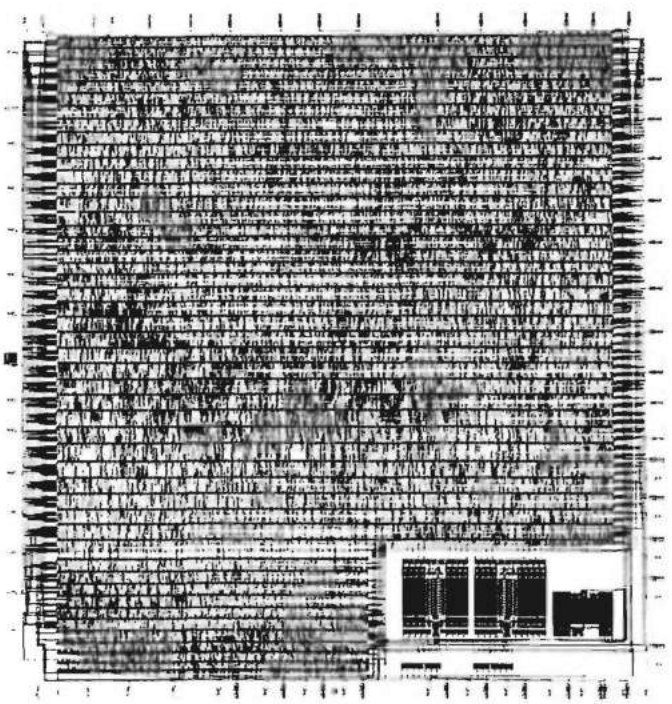
5.4.4.2.1.4 Circuitos especificados por el usuario o a medida. Al alcanzar el nivel de la VLSI los fabricantes de circuitos integrados pudieron ofrecer a los fabricantes de algunos equipos industriales (por ejemplo equipos de comunicaciones, electrodomésticos, etc.), la posibilidad de realizar un circuito específico para su aplicación. Estos circuitos se realizan a la medida del usuario (figura 5.28) y por ello reciben el nombre de CIDM a medida (en inglés «Custom integrated circuits»). Inicialmente se denominaron también CIDM especificados por el usuario [en inglés «User Specific Integrated Circuits (USIC)], pero dado que tienen una aplicación específica los fabricantes los denominan circuitos de aplicación específica [en inglés «Application Specific Integrated Circuits» (ASIC)].

Los circuitos integrados a medida han sido posibles gracias a la utilización de técnicas de especificación y de simulación asistidas por computador constituidas por programas ejecutados en potentes estaciones de trabajo (figura 5.29).

Para comprender los circuitos integrados a medida y sus diferentes variantes es conveniente analizar los distintos niveles de descripción de los sistemas digitales que se indican de forma resumida en la figura 5.30.



(a)



(b)

FIGURA 5.28. —**Circuito integrado a medida (custom) de muy gran escala de integración (VLSI) que constituye una unidad de acoplamiento de un procesador a una red local: *a*) descripción y diagrama de bloques; *b*) realización física** (Cortesía de Telefónica Investigación y Desarrollo).

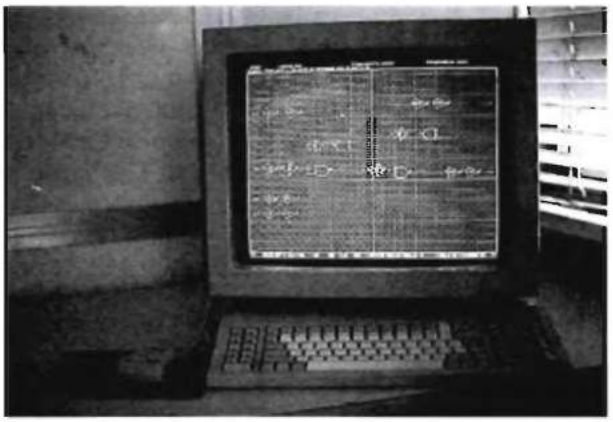


FIGURA 5.29. —**Estación de trabajo para el diseño de circuitos integrados semimedida (semi-custom) y totalmente a medida (full- custom)** (Cortesía del Departamento de Tecnología Electrónica de la Universidad de Vigo).

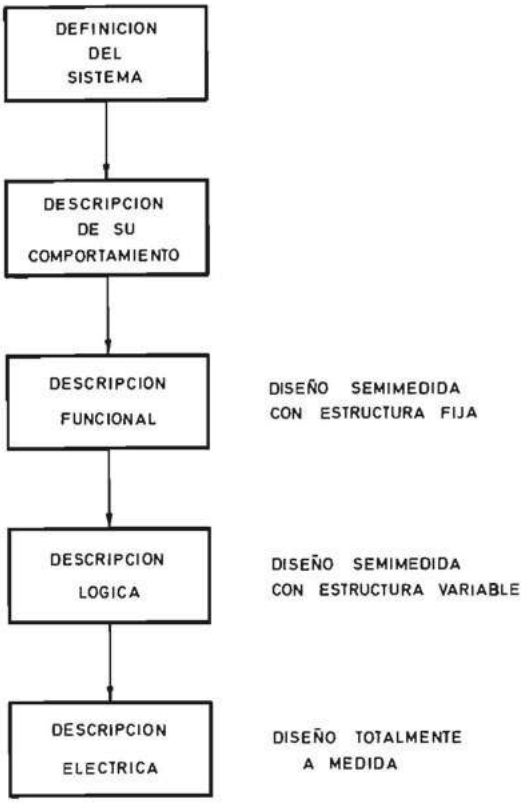


FIGURA 5.30. —**Niveles de descripción de un sistema digital y su relación con el diseño a medida.**

El nivel más alto está constituido por la descripción de su comportamiento, es decir, el conjunto de operaciones que constituyen el proceso que ha de realizar. A continuación se encuentra la descripción funcional o de bloques que incluye los distintos bloques funcionales y su descripción. Cada bloque está compuesto por un conjunto de elementos en cuya descripción se puede llegar hasta el nivel de la puerta lógica y dar lugar al nivel de descripción lógica. Finalmente se alcanza el nivel de la descripción eléctrica que especifica los componentes (transistores MOS por ejemplo), sus características y su interconexionado.

De acuerdo con el nivel hasta el que se llega en el diseño, los circuitos integrados a medida se clasifican en dos grandes tipos:

a) Circuitos integrados totalmente a medida (en inglés «Full-custom»).

Se definen como aquellos en los que el diseñador llega hasta el nivel de transistor, eligiendo sus características y el conexionado entre ellos. Permite obtener el máximo rendimiento a nivel de área de silicio, velocidad y potencia consumida pero en contrapartida presenta unos costes de diseño elevados. Por ello el diseño totalmente a medida sólo es aplicable cuando las series de fabricación son elevadas y se busca la optimización de las características y, en particular, de la velocidad de operación.

b) Circuitos integrados semimedida (en inglés «Semi-custom»).

Estos circuitos utilizan como bloques funcionales elementos predefinidos y por ello el diseñador adopta decisiones al nivel de descripción lógica sin alcanzar el de descripción eléctrica. Mediante la utilización de herramientas de diseño asistido por computador [HORB 86] [HURS 85], el diseñador utiliza elementos preespecificados denominados células (cells) y actúa a un nivel más alto de descripción que en el diseño totalmente a medida.

Existen diversas variantes de los circuitos integrados semimedida que se diferencian en la complejidad de las células utilizadas y la forma de interconectarlas. Entre ellas las más utilizadas son:

Los conjuntos de puertas programables mediante máscara [en inglés «Mask Programmable Gate Arrays» (MPGA)]

Consisten en un conjunto de elementos básicos dispuestos de forma regular y un conjunto de amplificadores de entrada y salida conectados a terminales externos. Su denominación indica que la función del circuito se determina mediante la selección de las máscaras adecuadas durante el proceso de fabricación.

Los elementos básicos pueden ser transistores de dimensiones y características predefinidas o puertas universales NO-Y (NAND) o NO-O (NOR) colocadas en posiciones fijas. Constituyen circuitos integrados semimedida porque, mediante la conexión de los transistores o puertas de una forma concreta, se convierten en circuitos que realizan una función específica.

En la figura 5.31 se representa una disposición típica de los elementos básicos de una MPGA. Los fabricantes de circuitos integrados ofrecen familias de MPGA

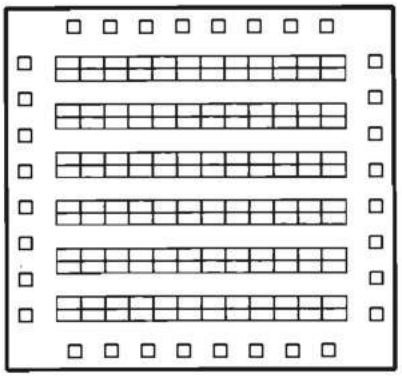


FIGURA 5.31.—**Disposición de los elementos lógicos de un circuito integrado monolítico semimedida basado en conjuntos de puertas (MPGA).**

con distinto número total de elementos y de terminales de entrada y salida, junto con las adecuadas herramientas de diseño asistido por computador. La mayoría de ellas incluyen la definición de macroceldas que corresponden al enlace de los transistores o puertas lógicas universales mediante unas determinadas conexiones para construir un cierto bloque funcional (como por ejemplo un multiplexor, un contador, etc.).

Las células estándar («Standard cells»)

Consisten en un conjunto o librería de células normalizadas que constituyen bloques funcionales de mayor complejidad que las puertas lógicas universales. La librería de células ha de contener un conjunto de funciones cuyo correcto funcionamiento esté garantizado: contadores, registros, decodificadores, etc. Pero la principal diferencia con las MPGA no radica en ello, porque en estas últimas se pueden definir macros como se indicó anteriormente, sino en que la colocación de las célu-

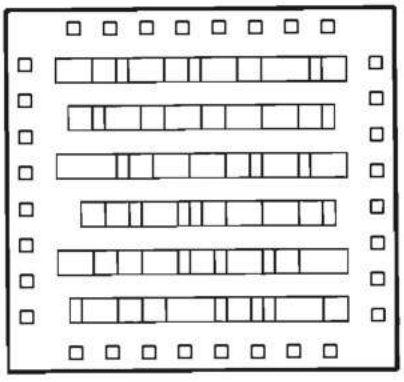


FIGURA 5.32.—**Disposición de los elementos de un circuito integrado monolítico semimedida basado en células estándar («Standard cells»).**

las y los canales para su interconexión no están predefinidos. Debido a ello las células estándar permiten un mejor aprovechamiento del silicio pero necesitan herramientas más potentes de diseño asistido por computador. En la figura 5.32 se representa una disposición típica de una célula estándar. La tendencia cada vez más acusada es la de disponer de complejos programas de diseño que reciben el nombre de compiladores de silicio («silicon compilers») que, a partir de la descripción del sistema en un lenguaje de alto nivel, deciden las células a utilizar, su colocación y conexionado, y establecen las diferentes máscaras que intervienen en el proceso de fabricación.

5.4.4.2.2 Circuitos integrados híbridos. Los circuitos integrados monolíticos son idóneos para la realización de sistemas digitales porque en ellos la precisión de los resultados no depende de la tolerancia de los componentes.

Pero en numerosas aplicaciones es necesario tratar digitalmente informaciones analógicas que deben convertirse a digitales por medio de convertidores analógico-digitales y sistemas de adquisición de datos en los que se combinan circuitos analógicos y digitales. En los circuitos analógicos se incluyen componentes pasivos como, por ejemplo, resistencias y condensadores de cuya precisión depende la de la conversión. Estos componentes necesitan un ajuste en la última fase del proceso de fabricación que resulta imposible de realizar con circuitos integrados monolíticos. De ahí el interés del desarrollo de los circuitos integrados híbridos que se presentan en dos versiones:

a) Circuitos híbridos de capa gruesa (Thick film circuits).

Se caracterizan por la deposición sobre un soporte cerámico de capas de espesor elevado (10 a 50 μm) de materiales resistivos, dieléctricos y conductores. Se logra así la realización de resistencias, condensadores y conductores que los interconectan entre sí y con circuitos integrados monolíticos y dispositivos transistores. Las resistencias pueden ser ajustadas mediante láser al final del proceso de fabricación lográndose así una elevada precisión. Se caracterizan por su bajo coste de fabricación, una menor densidad que los integrados monolíticos pero una gran fiabilidad y estabilidad a largo plazo de los componentes pasivos. Constituyen el 80 % de los circuitos integrados híbridos y se utilizan en circuitos analógico-digitales y para interconectar entre sí circuitos de gran (LSI) y muy gran (VLSI) escala de integración sustituyendo con ventaja a los circuitos impresos.

b) Circuitos híbridos de capa fina (Thin film integrated circuits).

Se diferencian de los de capa gruesa por el menor espesor de las capas que se depositan (inferior a algunas micras) que permiten obtener densidades mayores. Su proceso de fabricación es mucho más caro que el de los de capa gruesa y por ello su campo de aplicación es más reducido.

Un estudio más profundo de los circuitos integrados híbridos de capa gruesa (thick film hybrid integrated circuits) y de capa fina (thin film hybrid integrated circuits) se sale fuera de los límites de este libro y por ello se remite al lector a la bibliografía [GEIG 90] [SEID 83].

5.4.4.3 Circuitos integrados monolíticos digitales bipolares de silicio

5.4.4.3.1 Circuitos integrados de lógica resistencia-transistor (RTL). La primera tecnología diseñada en circuito integrado siguió la línea de lo realizado hasta ese instante con elementos discretos y circuitos bloque. En la figura 5.33 se representa el esquema de la puerta básica NO-O (NOR) en lógica positiva. El comportamiento de estas puertas desde un punto de vista lógico es igual al de la disposición representada en la figura 5.12.

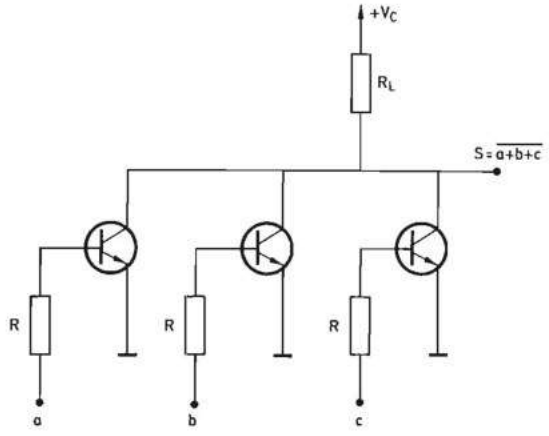


FIGURA 5.33.—**Tecnología RTL: Puerta NO-O (NOR) en lógica positiva.**

La diferencia tecnológica con respecto a aquélla es que cada entrada posee su propio transistor y la carga R_L es común a todos ellos. De esta forma las resistencias R y R_L se pueden dimensionar adecuadamente para que el transistor se sature (con un coeficiente de saturación $K_S = 1$) y se evita el problema de sobresaturación que se explicó sucedía en el esquema de la figura 5.12.

Cuando a cualquiera de las entradas se aplica una tensión positiva con respecto a masa (uno en lógica positiva), el transistor correspondiente conduce y la salida se pone aproximadamente a la tensión de masa (cero en lógica positiva). Por tanto, la puerta realiza la función NO-O en lógica positiva. El lector puede comprobar que en lógica negativa esta puerta realiza la función NO-Y (NAND).

Una de las principales diferencias entre este tipo de lógica y las que se estudian a continuación es que una puerta suministra corriente a las que se conectan a su salida y la limitación del número máximo de éstas viene dada por la máxima caída de tensión admisible en la resistencia de carga R_L, para que la tensión de salida, cuando ningún transistor conduce, lleve a saturación a todos los conectados a ella.

La máxima frecuencia de funcionamiento alcanzada en esta tecnología fue aproximadamente 5 MHz.

5.4.4.3.2 Circuitos integrados de lógica diodo-transistor (DTL). En la figura 5.34 se representa el esquema de la puerta básica de esta tecnología. Está formada

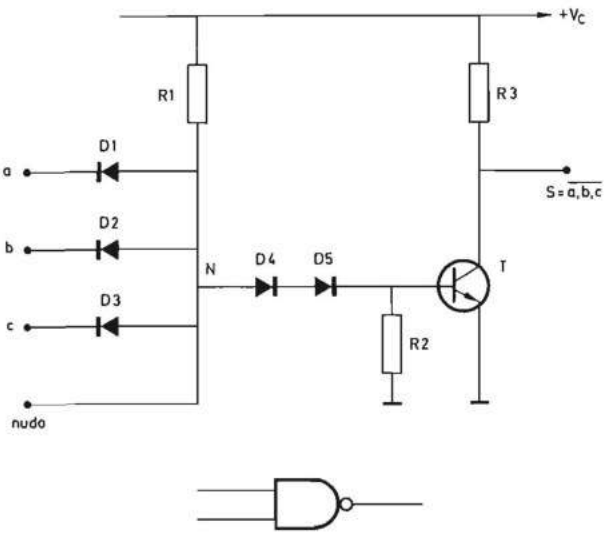


FIGURA 5.34.—**Tecnología DTL: Puerta NO-Y (NAND) en lógica positiva.**

por una puerta Y (en lógica positiva) constituida por los diodos D_1 y D_2 y D_3 y la resistencia R_1 y un circuito inversor realizado mediante el transistor T y las resistencias R_2 y R_3. El punto común a los tres diodos es accesible a fin de poder conectar exteriormente diodos y aumentar la cargabilidad de entrada del circuito.

Cuando cualquiera de las entradas de la puerta se conecta a la salida de otra puerta igual a ella cuyo transistor está saturado, a través del diodo correspondiente y R_1 circula una corriente, y la tensión en el punto N es aproximadamente igual a la caída de tensión directa en el diodo más la V_{CE} de saturación, aproximadamente 1 voltio en total. Los diodos D_4 y D_5 crean un umbral de tensión el cual impide que, en las condiciones citadas, el transistor T reciba corriente de base, por lo que la tensión de su colector es casi igual a la tensión de alimentación $+V_C$. Cuando todas las entradas de una puerta están conectadas a una tensión $+V_C$ directamente o a través de la resistencia R_3 de otra puerta cuyo transistor de salida está en corte, circula una corriente a través de R_1, D_4, D_5 y la unión base-emisor del transistor T que lleva a éste a saturación y la tensión en S toma un valor igual a V_{CE} de saturación.

Asignando el convenio de lógica positiva se comprueba que la función realizada por esta puerta es la representada en la tabla 5.4, que corresponde a la función NO-Y (NAND).

Fácilmente se comprueba que, si se adopta el criterio de lógica negativa, esta puerta realiza la función NO-O (NOR).

El umbral creado por D_4 y D_5 permite lograr una mínima inmunidad al ruido en estado cero y en estado uno (lógica positiva) de la entrada de una puerta, igual a 0,4 V y un valor típico de 1,2 V.

En aquellas puertas en las que el punto común a los ánodos de D_1, D_2 y D_3

c	b	a	s
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

TABLA 5.4

es accesible, la máxima cargabilidad de entrada es muy elevada y supera por término medio el valor 15. La cargabilidad máxima de salida es 10.

Con esta tecnología se han logrado tiempos de propagación por puerta de 30 ns y biestables de una frecuencia típica de operación de 10 MHz.

El producto potencia-tiempo de propagación típico es de 210 mW-ns. Una característica importante de esta tecnología de circuitos integrados es la de permitir la conexión directa de las salidas de dos puertas formando el montaje llamado «Y por conexión» (fig. 5.35). Estudiando esta figura y la figura 5.34 se observa que en lógica positiva la función obtenida es

$$S = \overline{ab}\ \overline{cd} = \overline{ab + cd}$$

debido a que la salida es cero cuando todas las entradas de cualquiera de las puertas están a uno. En la figura 5.35*a* se representa el montaje «Y por conexión» con símbolos lógicos no normalizados; para indicar que la unión directa de las salidas de las puertas realiza la función Y, se encierra en el símbolo de la misma la conexión.

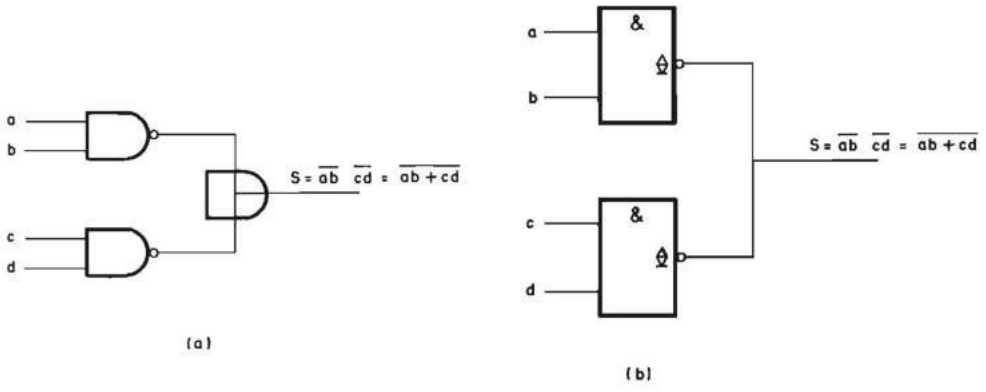


FIGURA 5.35.—**Montaje «Y por coxesión»: *a*) esquema con símbolos no normalizados; *b*) esquema con símbolos normalizados.**

En la figura 5.35*b* se representa el montaje «Y por conexión» con símbolos normalizados; en la salida de cada puerta se coloca el símbolo ⏚ para indicar que se trata de una salida en colector abierto con resistencia incorporada (ver apartado A1.3).

Si se conectan de esta forma puertas cuya resistencia del transistor de salida está incluida en el circuito integrado, se produce una reducción de la cargabilidad de salida en un número igual al de puertas conectadas. Para facilitar este montaje sin disminución de la cargabilidad de salida se realizaron puertas en circuitos integrados que no llevan la resistencia R_3 (fig. 5.36). La resistencia R_3 ha de elegirse para cada valor del número de puertas conectadas en esta forma N y la cargabilidad que se necesita M; R_3 no puede superar un cierto valor para que la tensión del uno lógico (lógica positiva) no descienda por debajo del mínimo valor admisible debido a las corrientes de pérdidas I_p (fig. 5.37) de las puertas que actúan de carga; por otro lado R_3 no puede descender por debajo de otro valor para que la corriente a través del transistor T de cualquier puerta del montaje «Y por conexión» no supere el máximo valor admisible. Los fabricantes proporcionaron unas tablas que definían el valor de R_3 para cada número N de puertas en el montaje «Y por conexión» y diferentes valores del número de puertas M que representa la cargabilidad de la conexión.

En la figura 5.37*a* se representa con símbolos no normalizados el montaje «Y por conexión» realizado con las puertas que carecen de resistencia de carga del transistor de salida, y que por ello reciben el nombre de puertas de colector abierto (open collector). En la figura 5.37*b* se representa el mismo montaje con símbolos normalizados; en la salida de cada puerta se coloca el símbolo ⍜ para indicar que se trata de una salida en colector abierto sin resistencia (ver apartado A1.3).

En el apartado 3.5.2 se estudian algunas aplicaciones del montaje «Y por conexión», que ha perdido interés práctico por las ventajas que presentan las puertas con carga activa que se estudian en el apartado siguiente. No obstante, el montaje «Y por conexión» sigue siendo el más adecuado para resolver algunas problemáticas de conexión de periféricos a un microprocesador y, en particular, el acopla-

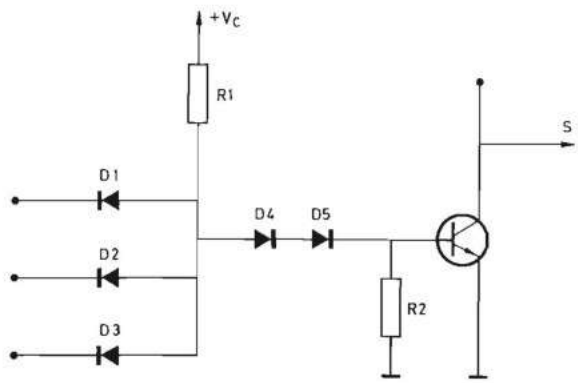


FIGURA 5.36.—**Tecnología DTL: Puerta NO-Y (NAND) con salida de colector abierto.**

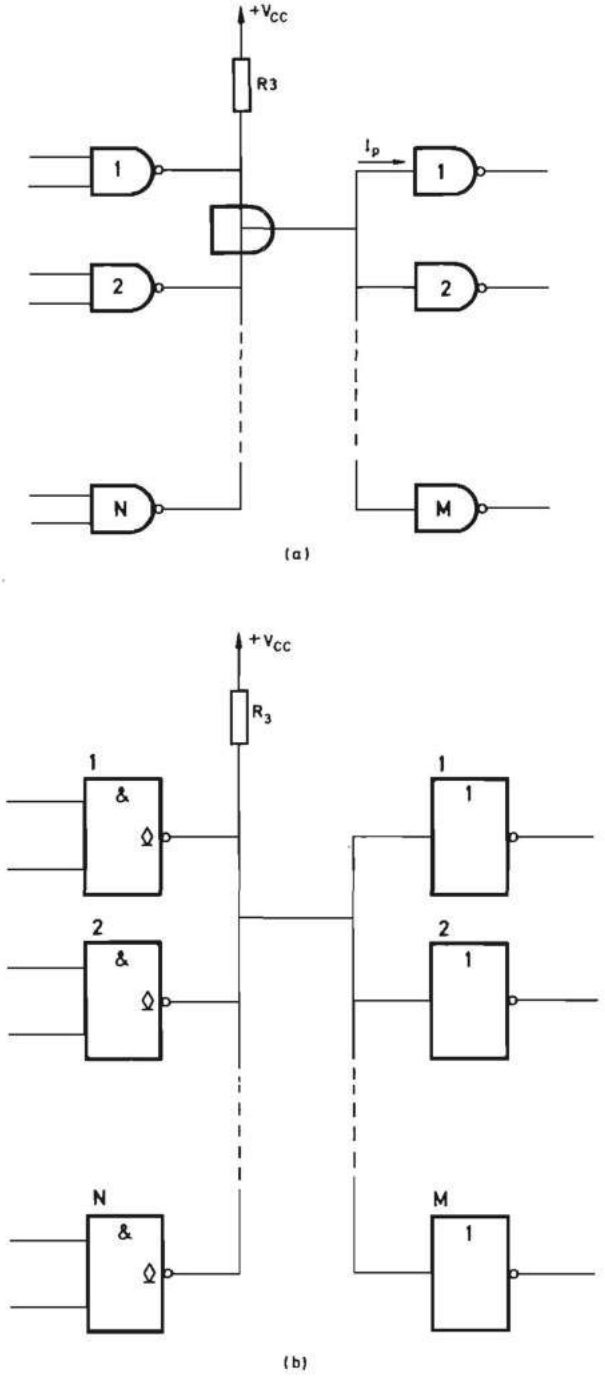


FIGURA 5.37.—**Montaje «Y por conexión» realizado con puertas de colector abierto: *a*) esquema con símbolos no normalizados; *b*) esquema con símbolos normalizados.**

miento por interrupción. Para su estudio se remite al lector a la bibliografía [MAND 80] [LIPO 88].

5.4.4.3.3 Circuitos integrados de lógica transistor-transistor (TTL). En la figura 5.38 se representa la puerta básica de esta tecnología que posee un conjunto de innovaciones con respecto a la tecnología diodo-transistor.

La puerta Y de entrada (lógica positiva) está constituida por la resistencia R_1 y un transistor multiemisor T_1 simétrico que realiza la misma misión que los diodos D_1 a D_3 de la figura 5.34.

A continuación de esta puerta existe un transistor en montaje divisor de fase formado por las resistencias R_2 y R_3 y el transistor T_2. Finalmente está la etapa de salida constituida por el transistor T_4 y su carga, formada por la resistencia R_4, el transistor T_3 y el diodo D_1.

En líneas generales el funcionamiento de esta puerta es el siguiente: Si cualquiera de los emisores del transistor T_1 se conecta a una tensión máxima de 0,8 V (tensión máxima V_{CE} del transistor de salida T_4 de una puerta cuando está saturado), se produce una corriente a través de R_1 y la unión base-emisor correspondiente. En estas condiciones el transistor T_2 está en corte y en consecuencia lo está también el transistor T_4; simultáneamente el transistor T_3 recibe una corriente de base a través de R_2, D_1 y la carga conectada a la salida S que normalmente será una o más puertas de tecnología TTL o bien una resistencia a masa (la corriente máxima que

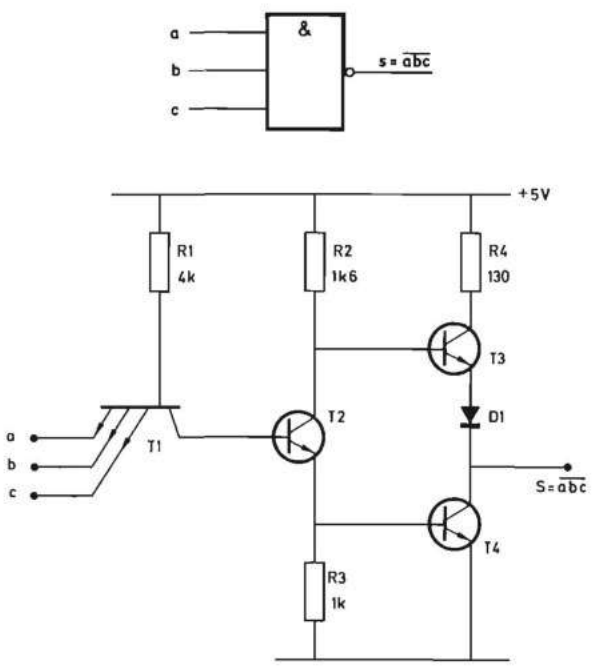


FIGURA 5.38.—**Tecnología TTL: Puerta NO-Y (NAND) en lógica positiva (tiempo de propagación 11 ns).**

puede salir de S hacia masa en 16 mA). Tal como se observa en la figura 5.38, cuando conduce T_3 la resistencia de carga es igual a R_4, es decir, 130 Ω. muy inferior a la resistencia de carga en tecnología DTL, que suele ser 2,2 kΩ. Esta es la causa principal de que el tiempo de propagación de la puertas TTL sea de 11 ns, porque la constante de tiempo RC de carga de los condensadores parásitos conectados a la salida de la puerta es muy reducida.

Cuando todos los emisores del transistor T_1 se conectan a una tensión superior a 2 V, se produce la conducción de la unión base-colector de este transistor y, en consecuencia, la saturación del transistor T_2 y el transistor T_4. En estas condiciones, la tensión en el colector de T_2 es insuficiente para hacer conducir a la unión base-emisor de T_3 debido a la existencia de D_1. Por tanto, en estas condiciones la resistencia de carga de T_4 es de varios megohmios. De lo dicho se deduce que la resistencia de carga de T_4 no es constante sino que vale aproximadamente 130 Ω cuando está en corte y varios megohmios cuando conduce. Esto permite reducir enormemente la potencia disipada y, como consecuencia, el producto potencia disipada-tiempo medio de propagación es tan sólo de 110 mW-ns. De todo lo dicho se deduce que, en régimen estático, los transistores T_3 y T_4 no conducen simultáneamente. Cuando el transistor T_2 cambia del estado de corte al de conducción, el transistor T_4 conduce antes de que deje de hacerlo T_3 y se produce un pico de corriente de duración muy pequeña (algunos ns) de aproximadamente 30 mA. Por el contrario, cuando T_2 pasa de saturación al corte, no se produce dicho pico de corriente debido a que T_4 se bloquea con cierta rapidez, por evacuarse su carga de base a través de R_3, y T_3 no conduce hasta que la tensión en su base haya llegado al valor necesario, para lo cual ha de cargarse su capacidad de difusión correspondiente.

De acuerdo con el convenio de lógica positiva o negativa adoptado, el lector puede comprobar que la puerta de la figura 5.38 realiza la función NO-Y o NO-O respectivamente.

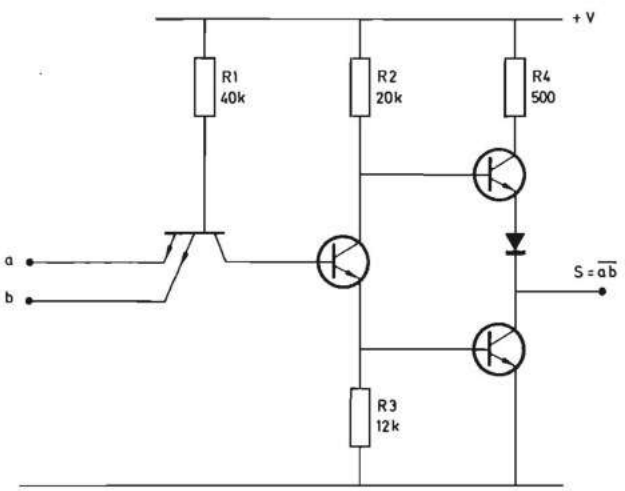


FIGURA 5.39.—**Tecnología TTL de baja potencia: Puerta NO-Y (NAND) en lógica positiva.**

Con la misma tecnología TTL descrita se han diseñado puertas de baja potencia elevando los valores de todas las resistencias tal como se indica en la figura 5.39. El producto potencia disipada-tiempo medio de propagación es igual al de la puerta normal y, por tanto, la disminución de potencia supone una elevación del tiempo de propagación.

Igualmente se han realizado puertas de tiempo de propagación medio de 6 ns reduciendo los valores de todas las resistencias (fig. 5.40) con lo cual se eleva la potencia disipada.

En tecnología TTL se han realizado también las puertas básicas representadas en la figura 5.41 en las que existen dos transistores divisores de fase en paralelo y dos transistores de entrada cuyo colector se conecta a la base de cada uno de aquéllos. El lector puede comprobar que asignando los criterios de lógica positiva y negativa esta puerta realiza las funciones NO-O y NO-Y respectivamente. Por tanto, en tecnología TTL, y adoptando un único criterio de lógica se dispone de puertas NO-Y y NO-O, lo cual representa una ventaja para realización de los sistemas digitales con un mínimo de elementos.

Los márgenes de ruido típicos de la tecnología TTL, son iguales a los de la tecnología DTL. Sin embargo, y como compensación al menor tiempo de conmutación, la inmunidad dinámica es más reducida.

El pequeño valor del producto potencia disipada-tiempo de propagación de esta tecnología ha motivado la realización de múltiples circuitos de escala de integración media que permiten diseñar sistemas digitales complejos con un número reducido de circuitos integrados, tal como se estudia en los capítulos 3 y 6.

Las puertas de tecnología TTL que acabamos de describir no permiten la realización del montaje «Y por conexión» porque la salida de una puerta no puede ser conectada directamente a masa a través de un transistor saturado, debido a que

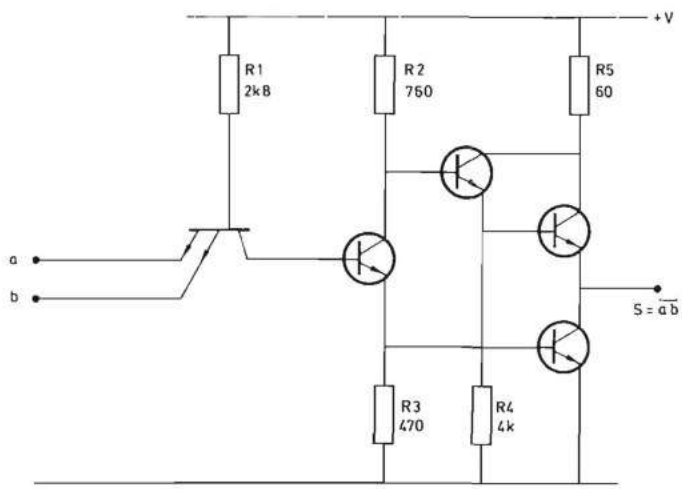


FIGURA 5.40.—**Tecnología TTL de gran velocidad: Puerta NO-Y (NAND) en lógica positiva (tiempo de propagación 6 ns)**

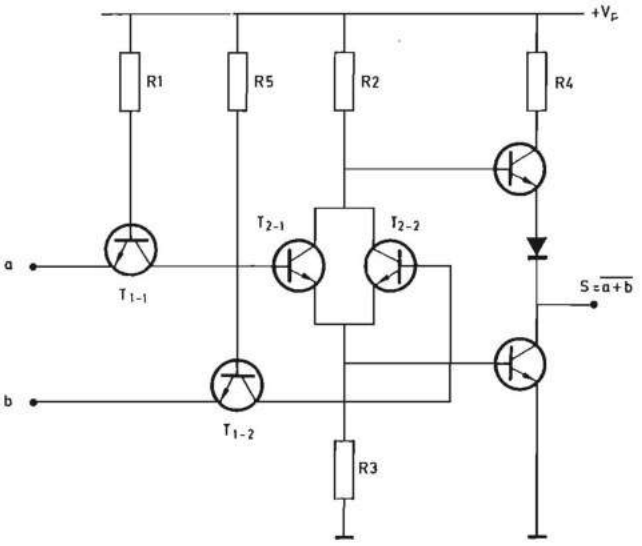


FIGURA 5.41.—**Tecnología TTL: Puerta NO-O (NOR) en lógica positiva.**

en dichas condiciones la corriente a través de T_3 excede el valor máximo admisible de 16 mA. Para poder realizar dicha conexión existen en tecnología TTL puertas en las que la carga del transistor de salida no está integrada, tal como se indica en la figura 5.42, y para las cuales son válidas las indicaciones dadas en el apartado 5.4.4.3.2. La gran desventaja del montaje «Y por conexión» en tecnología TTL es la elevación del tiempo de propagación, lo que era un inconveniente en DTL por ser esta tecnología mucho más lenta. Para solucionar este problema se ha modificado ligeramente la realización de la puerta básica TTL manteniendo todas las demás características de la misma y dando lugar a una nueva familia de circuitos integrados cuyas características se describen en el apartado siguiente.

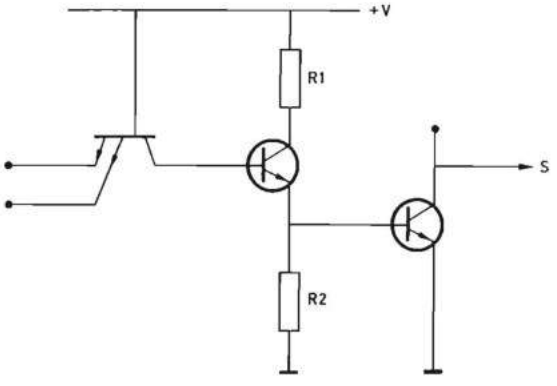


FIGURA 5.42.—**Tecnología TTL: Puerta NO-Y (NAND) en lógica positiva de colector abierto.**

5.4.4.3.4 Circuitos integrados de lógica transistor-transistor (TTL) de tres estados. La puerta básica de esta tecnología se representa en la figura 5.43. Se diferencia fundamentalmente de la puerta TTL representada en la figura 5.39 en que las salidas de dos puertas pueden ser unidas entre sí directamente. Para lograrlo se dispone de una entrada de inhibición I y el circuito formado por las resistencias R_6, R_7 y R_8, los transistores T_6, T_7 y T_8 y el diodo D_1. Cuando el nivel de tensión aplicada a I es superior a 2 V, se produce una corriente en la unión base-colector de T_6 que lleva a la saturación a los transistores T_7 y T_8. El colector del transistor T_8 está unido a un emisor de T_1 y a la base de T_3 a través del diodo D_1 y, por tanto, su saturación hace que los transistores T_4 y T_5 estén en corte independientemente del estado de las entradas a y b. En estas condiciones, la impedancia en-

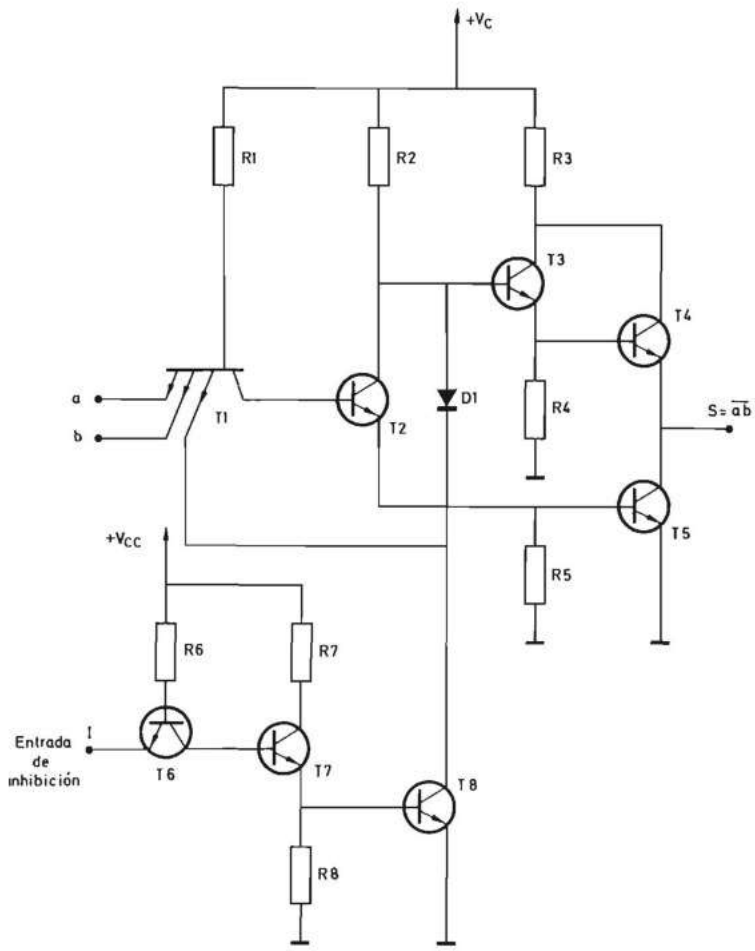


FIGURA 5.43.—**Tecnología TTL: Puerta NO-Y (NAND) en lógica positiva con salida de tres estados.**

tre la salida y la alimentación $+V_C$, al igual que la impedancia entre la salida y masa, son elevadas, del orden de varios megohmios. Este tercer estado de la puerta (que no existe en los circuitos TTL estudiados en el apartado anterior) es el que da nombre a esta tecnología.

Si la entrada *I* está a un nivel de tensión inferior a 0,8 V, se anula la corriente base-colector de T_6 y se lleva al corte a T_8, con lo que la puerta se comporta exactamente igual que la de la figura 5.38 y el estado de la salida *S* depende de las entradas *a* y *b*. El lector puede comprobar que la puerta realiza la función NO-Y en lógica positiva y la función NO-O en lógica negativa.

En la figura 5.44*a* se representa el esquema lógico correspondiente a la puerta de la figura 5.43 realizado con símbolos no normalizados, y en la figura 5.44*b* el símbolo abreviado utilizado en los esquemas.

En la figura 5.44*c* se representa la misma puerta mediante un símbolo lógico normalizado. La salida incorpora el símbolo ▽ para indicar que es de tres estados, y la puerta posee una entrada adicional de desinhibición (Enable) que provoca dicho estado cuando se encuentra en nivel cero.

En la figura 5.45 se representa un esquema práctico de montaje. Sólo una de las entradas de inhibición I_1 ($\overline{EN}1$) a I_n ($\overline{EN}n$) puede estar en estado cero en cualquier instante; en esta situación el estado de la conexión de todas las salidas de las puertas depende exclusivamente del estado de las entradas de la puerta cuya entrada *I* ($\overline{EN}$) se encuentra en dicho estado cero. La máxima corriente a través de T_4 (fig. 5.43) para que el nivel del uno lógico no descienda por debajo del máximo admisible es 5,2 mA y la corriente de pérdidas a través de T_5 en corte y de una entrada del transistor multiemisor T_1 es de 40 μA. Así, pues, para tener una cargabilidad de salida de 3 unidades (120 μA), el máximo número *n* de puertas que se pueden conectar entre sí viene dado por la expresión:

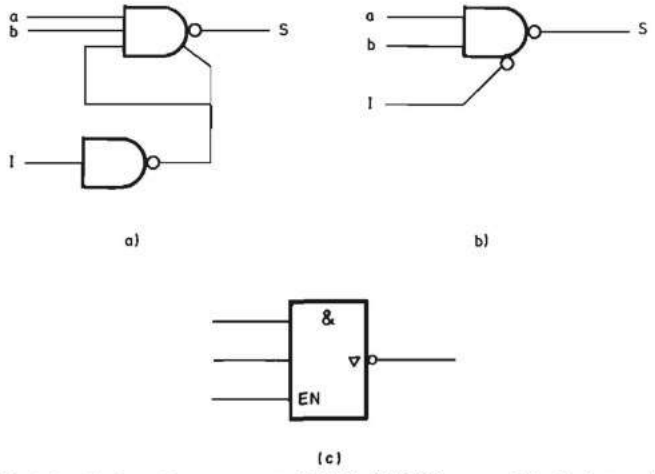
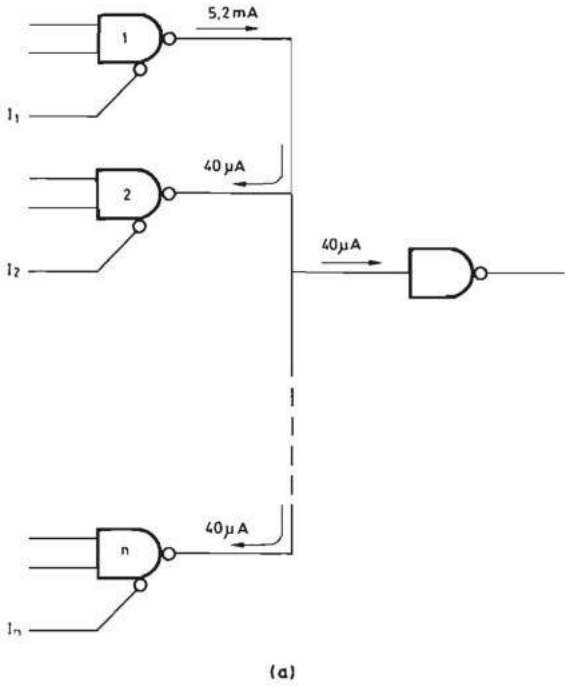
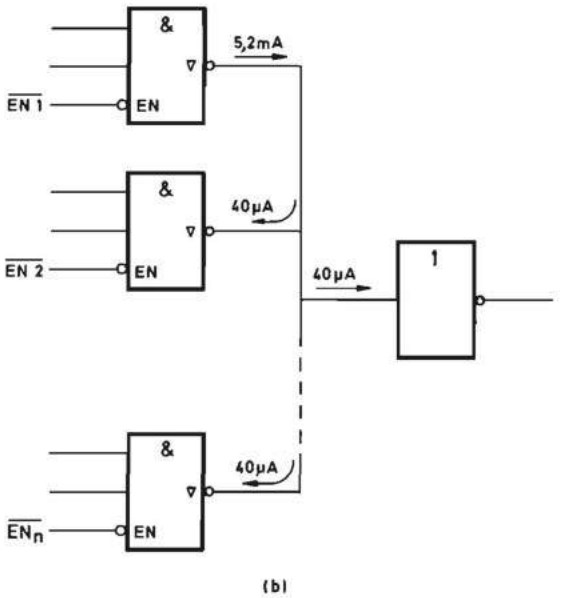


FIGURA 5.44.—**Símbolos lógicos de una puerta NO-Y (NAND) con salida de tres estados: *a*) no normalizado completo; *b*) no normalizado abreviado; *c*) normalizado.**



(a)



(b)

FIGURA 5.45.—**Montaje con puertas NO-Y (NAND) con salida de tres estados.**

$$5200\ \mu A - 120\ \mu A = (n - 1)\ 40\ \mu A$$

$$n - 1 = \frac{5080}{40}$$

de donde resulta:

$$n = 128$$

Una de las principales aplicaciones de la tecnología TTL de tres estados es la realización de circuitos de escala de integración media constituidos por puertas TTL normales, excepto las de salida, que son de tres estados y por tanto pueden ser conectadas entre sí. Esto permite enviar a una conexión común denominada barra (bus) la información procedente de un cierto número *n* de circuitos, seleccionando en cada instante el circuito que la envía.

El concepto de barra (bus), posible en la práctica gracias a las tecnologías de tres estados, es uno de los más importantes de las técnicas digitales. Sus aplicaciones se describen, entre otros, en el apartado 6.3.5.4.6 (conjuntos de registros) y 7.2.3.1.4 (unidades de memoria).

5.4.4.3.5 Tecnología de alta inmunidad al ruido. Las tecnologías DTL y TTL estudiadas anteriormente presentan una inmunidad al ruido relativamente baja y en ellas no es posible realizar temporizadores como el de la figura 5.13 de un valor elevado de *RC*.

Ello propició el desarrollo de una nueva tecnología que presentó una inmunidad típica de 5 V y por ello recibió el nombre de alta inmunidad al ruido. Esta tecnología permitió la integración de temporizadores de elevada constante de tiempo y constituyó en su momento (1970 a 1973) una solución adecuada para el diseño de sistemas electrónicos de control industrial de reducida complejidad que tuviesen que trabajar en ambientes con niveles de ruido eléctrico elevado.

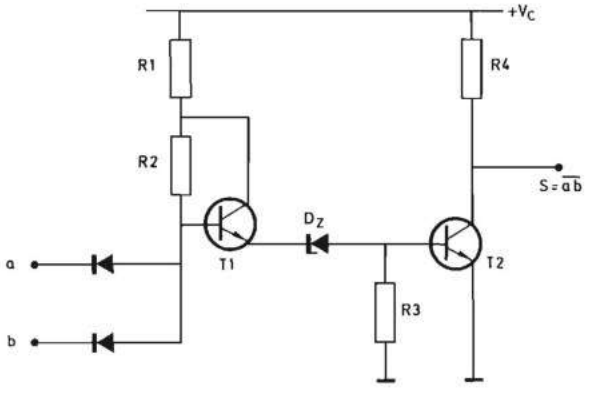


FIGURA 5.46.—**Puerta NO-Y (NAND) básica en lógica positiva de alta inmunidad al ruido.**

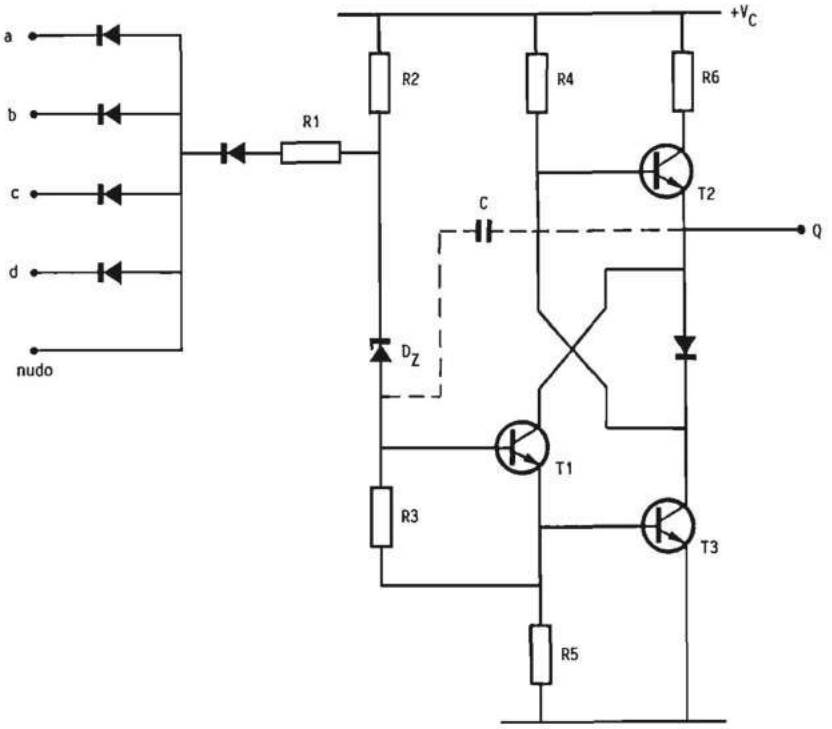


FIGURA 5.47.—**Tecnología de alta inmunidad al ruido (HTL): Puerta NO-Y (NAND) en lógica posiva.**

En la figura 5.46 se representa el esquema fundamental de una puerta de tecnología HTL que realiza la función NO-Y en lógica positiva. Básicamente es igual a la puerta DTL representada en la figura 5.34 sustituyendo al diodo D_1 de aquélla por el transistor T_1 y el diodo D_2 por el diodo zener D_z, el cual proporciona la elevada inmunidad al ruido. En efecto, si ambas entradas *a* y *b* están conectadas a las salidas de sendas puertas cuyo transistor T_2 correspondiente está saturado, para que un ruido presente en cualquiera de ellas haga cambiar el estado de la salida es necesario que supere el valor de la tensión de zener, aproximadamente 5 V. La tensión de alimentación de las puertas de esta tecnología es alrededor de los 15 V, muy superiores a la de las demás tecnologías bipolares.

En la figura 5.47 se representa el esquema de una puerta más evolucionada de alta inmunidad al ruido cuya principal característica es la de tener accesible el cátodo del diodo zener D_z. Conectando un condensador entre el punto Q y la salida se puede elevar el tiempo de propagación de la puerta y, por tanto, incrementar la inmunidad dinámica al ruido tal como se indica en la figura 5.48. De esta forma el diseñador de sistemas lógicos puede adaptar el tiempo de propagación medio de las puertas a la frecuencia de operación del montaje concreto. La máxima frecuencia de esta tecnología es 500 kHz.

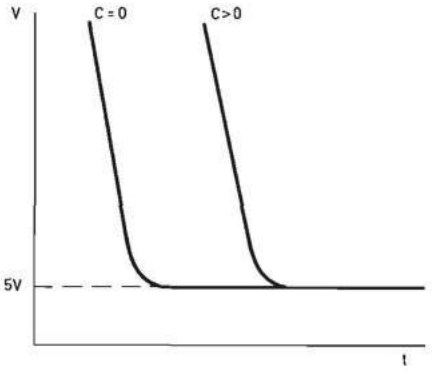


FIGURA 5.48.—**Curvas de inmunidad dinámica al ruido de la tecnología de alta inmunidad al ruido (HTL).**

La principal desventaja de esta tecnología fue su bajo nivel de integración debido a las dificultades de realización de los diodos zener. Por ello fue rápidamente desplazada por las tecnologías con transistores MOS y en especial la CMOS que se estudia en el apartado 5.4.4.4.2.3. En la actualidad la tecnología de alta inmunidad al ruido no se utiliza en el diseño de sistemas digitales.

5.4.4.3.6 Circuitos digitales de lógica no saturada. Los circuitos integrados bipolares en que los transistores llegan al estado de saturación tienen un límite mínimo de tiempo de propagación determinado por el tiempo de almacenamiento de los portadores de carga en la unión base-colector. En muchos procesadores digitales que trabajan en tiempo real es necesario reducir al máximo el tiempo de propagación y esto ha motivado la aparición de los circuitos de lógica no saturada.

Para anular el tiempo de almacenamiento se han diseñado varios circuitos que han dado lugar a otras tantas tecnologías, de entre las cuales se estudian a continuación las de mayor difusión.

5.4.4.3.6.1 Circuitos integrados de tecnología TTL Schottky. La solución empleada en esta tecnología para evitar la saturación de los transistores consiste en conectar, en paralelo con su unión base-colector, un diodo (su cátodo se une al colector por ser el transistor *NPN*) cuya caída de tensión directa es menor que la de la propia unión.

El diodo utilizado es del tipo Schottky, el cual está constituido por una unión de aluminio con un semiconductor y tiene la propiedad de que su caída de tensión directa es inferior a la de una unión *PN* normal en 200 o 300 mV. Este diodo impide que la unión base-colector se polarice en sentido directo, con lo cual se evita la saturación del transistor.

FIGURA 5.49.—**Símbolo de un diodo Schottky.**

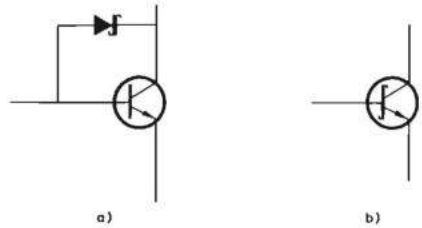


FIGURA 5.50.—**Montaje de un diodo Schottky y un transistor bipolar: *a*) esquema del circuito; *b*) símbolo abreviado.**

En la figura 5.49 se indica el símbolo representativo de un diodo Schottky. En la figura 5.50*a* se representa el montaje de un diodo Schottky y un transistor bipolar de silicio *NPN* y en la figura 5.50*b* el símbolo abreviado de este montaje.

La puerta básica de esta tecnología se representa en la figura 5.51*a* y, como podemos comprobar, es muy similar a la puerta de la tecnología TTL representada en la figura 5.51*b*. Todos los transistores son del tipo Schottky excepto T_4 que

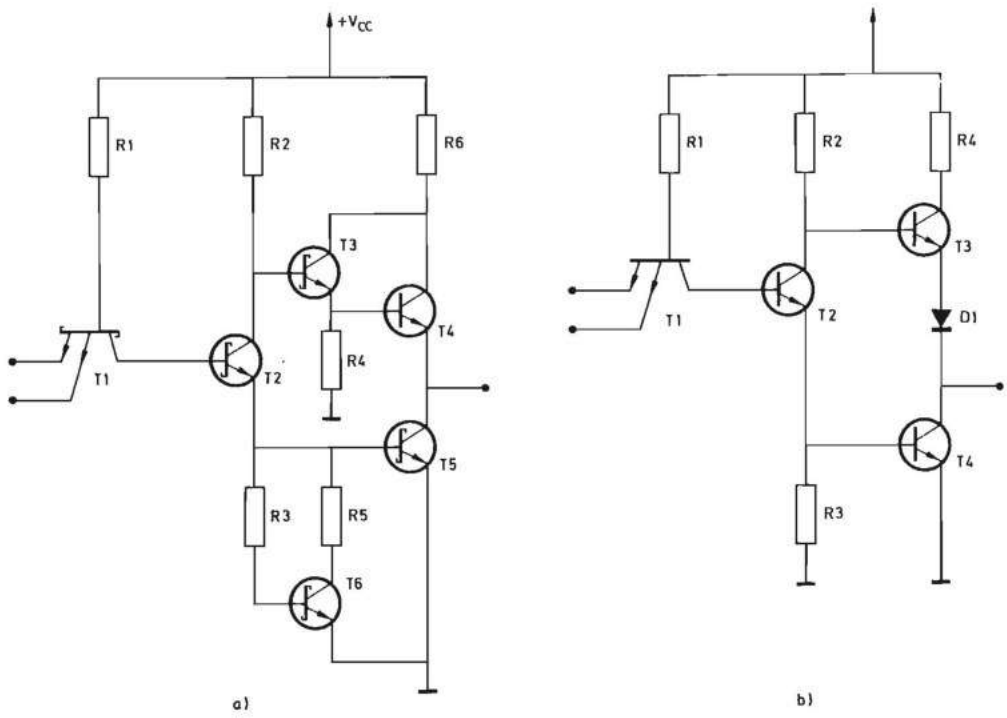


FIGURA 5.51.—*a*) **Puerta NO-Y (NAND) en lógica positiva de tecnología TTL Schottky; *b*) puerta NO-Y (NAND) de tecnología TTL.**

nunca llega a la saturación porque a través de él solamente circulan las corrientes de pérdidas de las puertas de la misma tecnología conectadas a la salida, cuando T_5 está al corte. El montaje formado por los transistores T_3 y T_4 (fig. 5.51*a*), es estáticamente equivalente al formado por el transistor T_3 y el diodo D_1 de la figura 5.51*b*, pero con una mejor respuesta dinámica. El transistor T_6 aumenta la rapidez de conmutación del transistor T_5.

Por todo lo expuesto, esta tecnología recibe el nombre de TTL Schottky y posee un menor producto de la potencia consumida por el tiempo de propagación que la TTL normal.

La puerta de la figura 5.51*a* realiza la función NO-Y (NAND) en lógica positiva y la NO-O (NOR) en lógica negativa. Las primeras puertas de tecnología TTL Schottky poseían un tiempo medio de propagación de 3 ns y una potencia media disipada de 20 mW. Al igual que en tecnología TTL normal, se ha desarrollado una tecnología TTL Schottky de baja potencia (Low power TTL Schottky) y a través de diversas modificaciones en los circuitos se han logrado las tecnologías TTL Schottky avanzada (Advanced Schottky) y TTL Schottky avanzada de baja potencia (Advanced low power Schottky) que presentan un menor producto de la potencia disipada por el tiempo de propagación [TEXA 84].

5.4.4.3.6.2 Circuitos integrados de lógica de acoplamiento por emisor (ECL). En la figura 5.52 se representa el esquema básico de un circuito de lógica de acoplamiento por emisor en el cual se observa que existen dos tensiones de alimentación diferentes: V_E y V_{CC} ($|V_E| < |V_{CC}|$). Si a la entrada *e* se aplica una tensión más negativa que V_E, el transistor T_1, no conducirá y lo hará T_2.

Si, por el contrario, la tensión en *e* es menos negativa que V_E, conducirá T_1 y no lo hará T_2.

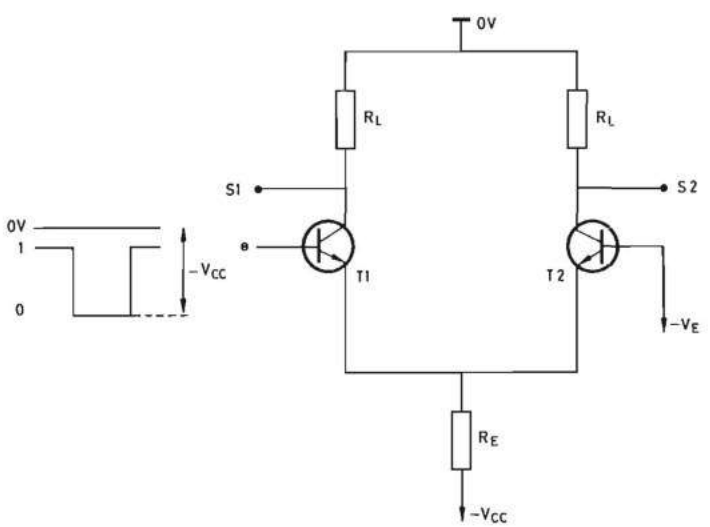


FIGURA 5.52.—**Circuito básico de lógica de acoplamiento por emisor [emitter coupled logic (ECL)].**

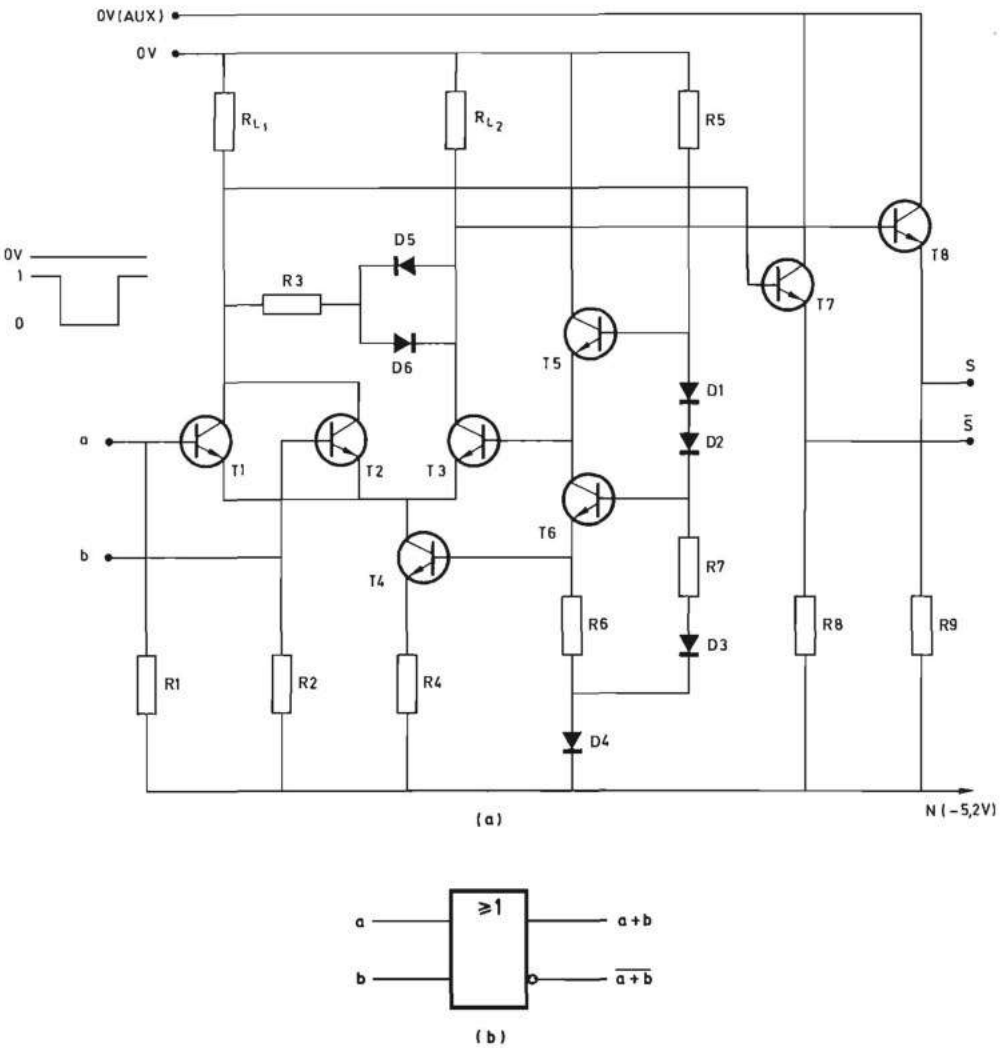


FIGURA 5.53.—**Tecnología de acoplamiento por emisor (ECL): *a*) esquema de una puerta O y NO-O en lógica positiva; *b*) símbolo lógico normalizado.**

Si $R_L I_C \leq |V_E|$ los transistores T_1 y T_2 no se saturarán y de esta forma se puede lograr un tiempo medio de propagación de 1 a 2 ns. Se observa que la salida S_2 está en fase con la entrada e y que S_1 está en oposición de fase.

Los diversos fabricantes han adoptado diferentes configuraciones para la realización de estos circuitos. En general, la tensión $-V_E$ se genera a partir de $-V_{CC}$ en el propio circuito.

En la figura 5.53*a* se representa una puerta de lógica no saturada. El transistor T_4 actúa como fuente de corriente y T_5 establece la tensión de referencia de la

base de T_3. Los colectores de los transistores T_1, T_2 y T_3 se conectan a sendos transistores T_7 y T_8 en seguidor de emisor, lo que permite obtener una cargabilidad de salida elevada y unos niveles de salida compatibles con la entrada de otra puerta de idénticas características.

El terminal N se alimenta negativamente con respecto a la masa. Los dos niveles lógicos de las entradas son negativos con respecto a masa. Si ambas entradas toman el nivel más negativo, no conducirán T_1 ni T_2 y lo hará T_3 sin llegar a saturarse, con lo cual su colector se hará negativo. Este nivel se reflejará a la salida S, que es el emisor de T_8, el cual está cercano al corte. Por otra parte, el colector de T_1 estará al nivel de masa, lo cual hace conducir a T_7 y, por tanto, la salida $\overline{S}$ estará aproximadamente al nivel de masa (1 en lógica positiva). Por el contrario, si cualquier entrada a o b se encuentra en el nivel menos negativo, conducirán respectivamente T_1 o T_2 y no lo hará T_3, con lo cual ambas salidas cambian de estado. Adoptando el convenio de lógica positiva, esta puerta realiza las funciones O y NO-O en sus salidas respectivas. El resto del circuito genera la tensión de referencia y compensa las variaciones con la temperatura de la tensión de referencia y la tensión de salida. En la figura 5.53*b* se representa el símbolo lógico normalizado de esta puerta.

Una desventaja de esta tecnología es la reducida inmunidad al ruido que obliga a incrementar las precauciones para anular la influencia del ruido generado por el propio sistema y el proveniente de fuentes externas de ruido. Precisamente, para evitar la influencia de las caídas de tensión en los hilos de alimentación, se han independizado los terminales de alimentación de los transistores de salida montados en seguidor de emisor y los transistores de entrada T_1, T_2 y T_3 (fig. 5.53*a*).

5.4.4.3.7 Tecnología de inyección integrada (Integrated Injection Logic [IIL o I^2L]). La búsqueda de una mayor densidad de integración llevó, al principio de la década de los setenta, al desarrollo de la tecnología de inyección integrada (I^2L).

Esta tecnología constituye una evolución de los circuitos lógicos realizados con transistores acoplados directamente [Direct coupled transistor logic (DCTL)] que se desarrollaron a principios de la década de los sesenta. En la figura 5.54*a* se representa el esquema de un conjunto de puerta NO-O (NOR) de tecnología DCTL y en la 5.54*b* su esquema lógico equivalente. La puerta utilizada en este montaje es de tres entradas, y se puede comprobar fácilmente su funcionamiento. Si todas las entradas están a masa («0» en lógica positiva), los transistores están al corte y dado que la salida se conecta a la base de otro u otros transistores, el nivel es aproximadamente 0,7 V. Si cualquiera de los transistores tiene en su base una tensión mayor de 0,7 V pasa a saturación y el nivel de su salida es aproximadamente de 0,1 a 0,2 V. Por lo tanto, la separación entre los dos niveles lógicos es de 0,5 a 0,6 V.

El principal inconveniente de la tecnología DCTL es el debido a la imposibilidad de lograr transistores de características idénticas, en especial en su comportamiento con la temperatura. Si los transistores cuya unión base-emisor está en paralelo (por ejemplo, *T*1 y *T*2 en la figura 5.54*a*) necesitan distintas tensiones en dicha unión para alcanzar la saturación, el que necesita una tensión menor es el único

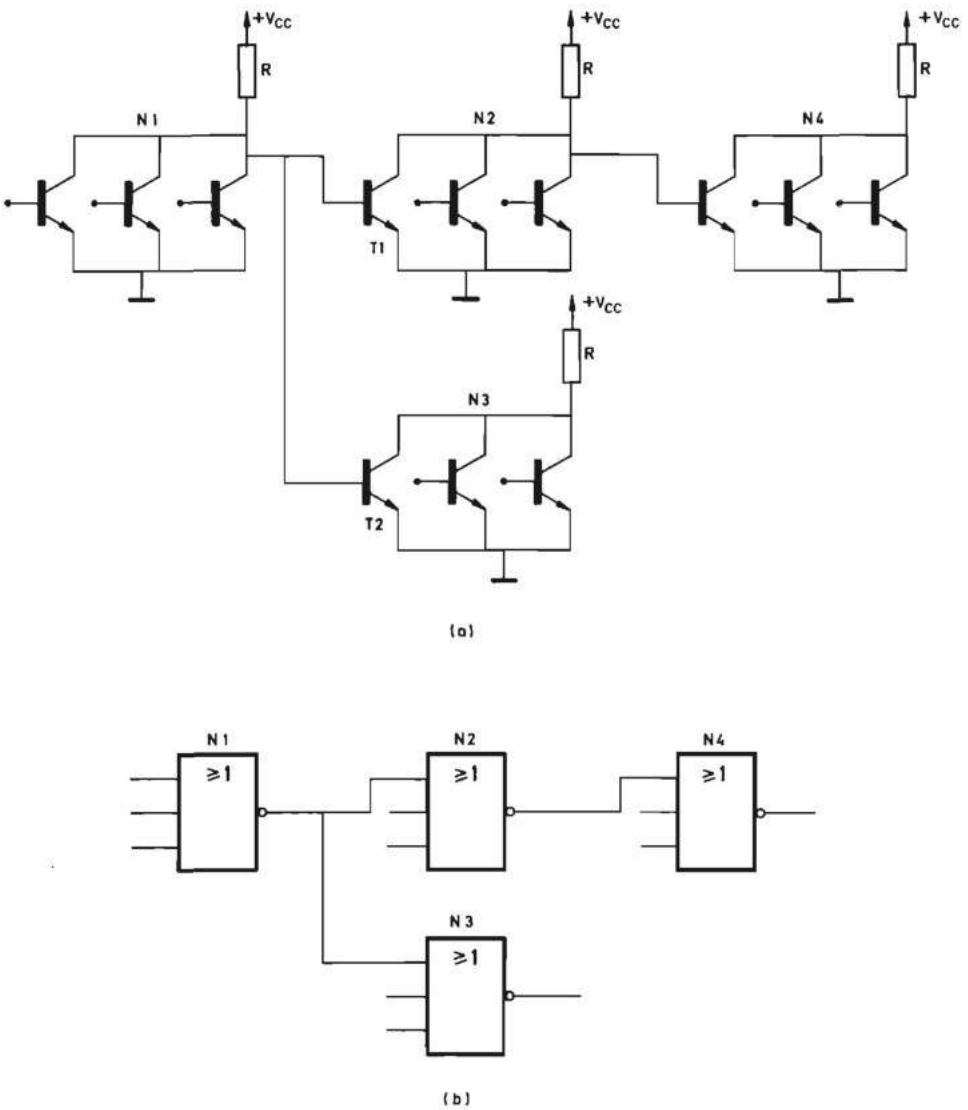


FIGURA 5.54. —**Esquema de un conjunto de puertas NO-O (NOR) en tecnología de transistores directamente acoplados [Direct coupled transistor logic (DCTL)]: *a*) esquema del circuito; *b*) esquema lógico equivalente.**

que alcanza dicha situación y tiene una tensión colector-emisor de 0,1 a 0,2 V. Por el contrario, el que necesita una tensión colector-emisor mayor no se satura y tiene una tensión colector-emisor superior a la anteriormente indicada. Para ilustrar este fenómeno, que recibe el nombre de acaparamiento de corriente (hogging), se representa en la figura 5.55 el circuito equivalente a la unión base-emisor de los transistores *T*1 y *T*2. Se supone que *T*1 necesita 0,7 V en la base para saturarse

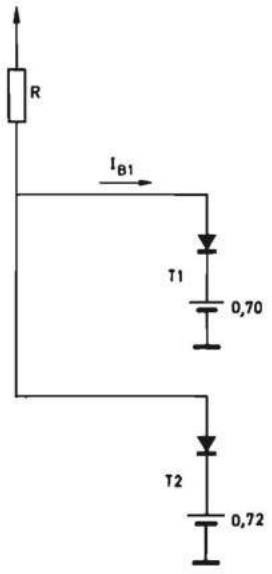


FIGURA 5.55.—**Circuito equivalente a la unión base-emisor de los transistores *T*1 y *T*2 de la figura 5.54.**

y, por ello, su corriente de base I_{B1} es igual a $\frac{V_{CC}-0,7}{R}$; por el contrario *T*2 necesita 0,72 V y, por lo tanto, $I_{B2} \ll I_{B1}$. Este fenómeno disminuye el margen de ruido porque la tensión de salida de la puerta cuyo transistor no se satura es superior a 0,2 V. Por todo ello esta tecnología fue abandonada y sustituida por la RTL que a su vez fue seguida por las DTL y TTL.

La tecnología TTL ya estudiada alcanzó una gran madurez que la convirtió en un estándar del mercado en sus diferentes versiones, en especial las que incorporan los diodos Schottky. Pero la gran cantidad de resistencias que incorpora una puerta TTL, que ocupan mayor superficie que los transistores, limitó la densidad de integración para poder alcanzar circuitos de muy gran escala de integración (VLSI).

Por ello se desarrolló la tecnología I^2L que conserva la simplicidad de la DCTL al utilizar una sola resistencia por puerta y evitar el inconveniente citado al analizar aquélla. La tecnología I^2L se basa en la sustitución de los transistores de la DCTL que poseen la unión base-emisor en paralelo, por un único transistor multicolector.

Para ilustrarlo gráficamente, en la figura 5.56 se representa un circuito DCTL que obtiene los cuatro productos canónicos de dos variables A y B. Analizando este circuito se observa que en los transistores *T*1, *T*3 y *T*5 se presenta el problema de acaparamiento (hogging) de corriente. Pero el circuito de la figura 5.56 es equivalente al representado en la figura 5.57 en el que se agrupan los transistores cuyas uniones base-emisor están conectadas en paralelo. Si dichos transistores se sustitu-

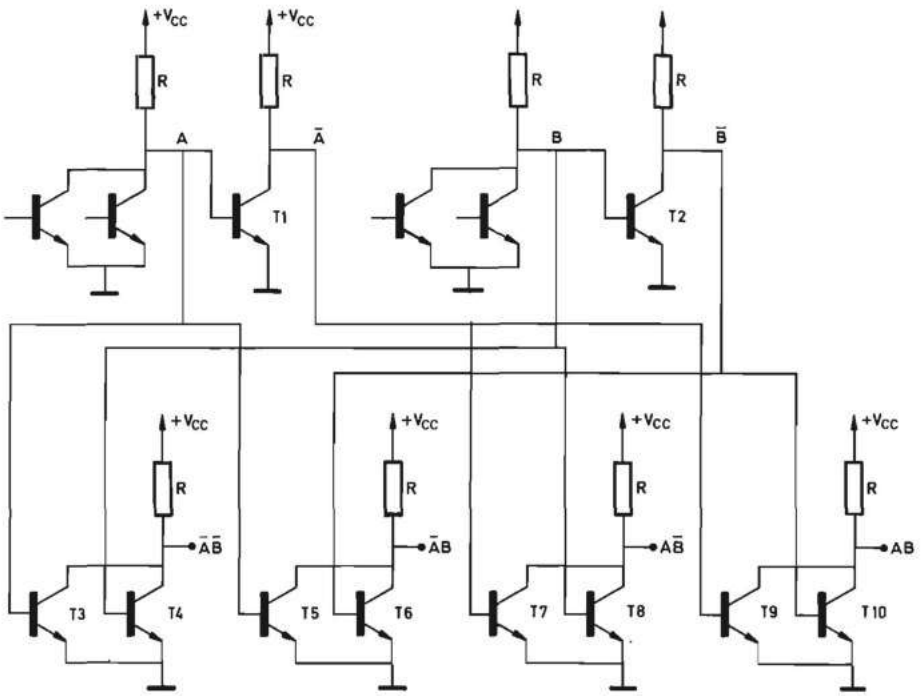


FIGURA 5.56.—**Circuito decodificador de los cuatro productos canónicos de dos variables en tecnología de transistores directamente acoplados (DCTL).**

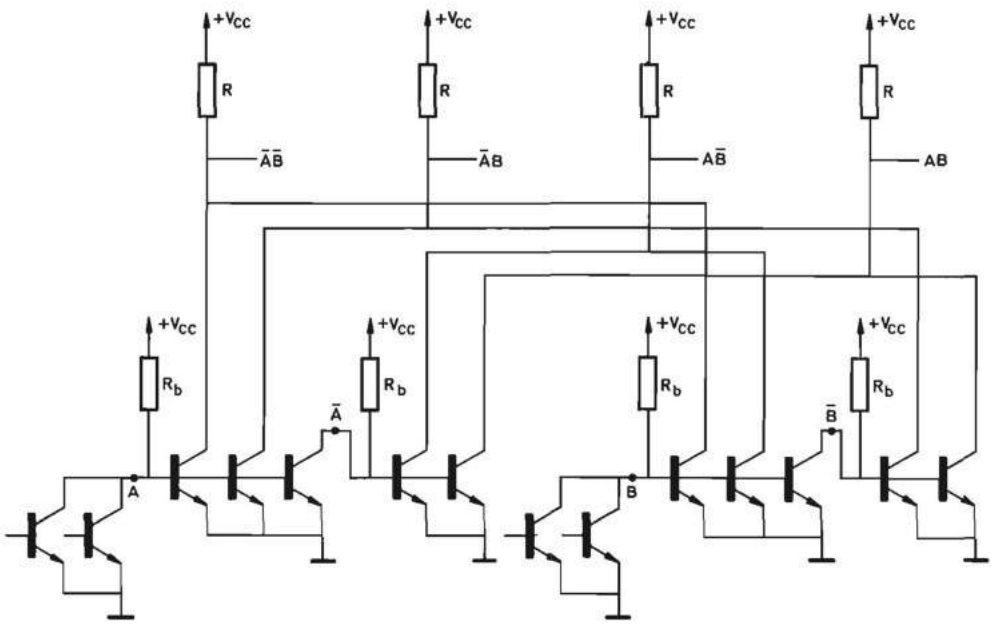


FIGURA 5.57.—**Circuito equivalente al de la figura 5.56.**

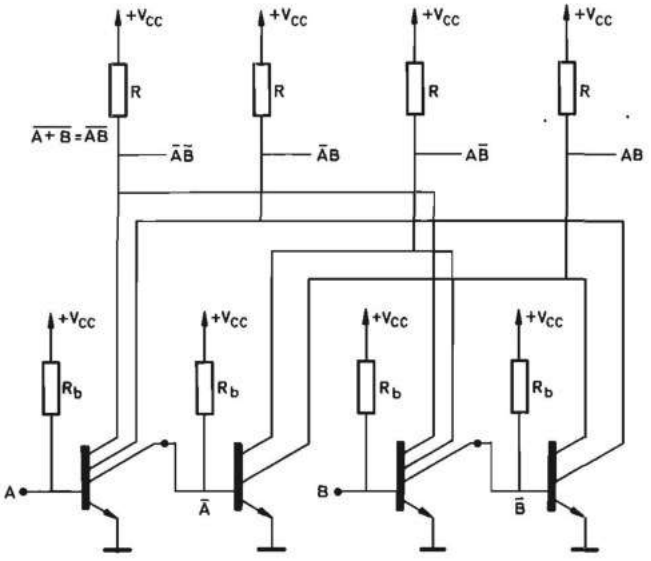


FIGURA 5.58.—**Circuito equivalente al de la figura 5.57 realizado con transistores multicolector.**

yen por uno multicolector, se obtiene el esquema de la figura 5.58 que constituye el circuito básico de las puertas de lógica de inyección integrada (IIL o I^2L) en el que se evita el fenómeno de acaparamiento por no existir varias uniones base-emisor en paralelo. Además, para reducir el número de resistencias y elevar la densidad de integración, se sustituyen las resistencias R_b por transistores *PNP* (figura 5.59) con la base conectada a masa. Estos transistores inyectan corriente en la base de los transistores conmutadores y de ahí viene el nombre de lógica de inyección integrada.

Los niveles de tensión que aparecen en los colectores de los transistores conmutadores varían de 0,7 V (estado de corte) a 0,1 o 0,2 V (estado de saturación), lo cual supone un margen de ruido muy pequeño. Esto se solventa no conectando a una unión base-emisor los colectores de salida de los transistores conmutadores, con lo cual la diferencia entre los niveles es prácticamente igual a la tensión de alimentación. Es usual la realización de circuitos integrados con tecnología I^2L en su interior y etapas de acoplamiento a la entrada y a la salida compatibles con tecnología TTL para lograr circuitos VLSI que se pueden conectar a los ya existentes en esta última tecnología.

Mediante la elección adecuada del valor de R_x se puede hacer que los transistores conmutadores trabajen en régimen de mayor o menor saturación y controlar de esta forma el tiempo de propagación de las puertas.

Una ventaja adicional de la tecnología I^2L es la reducida corriente a través de los transistores (típicamente de 1 a 10 μA), lo cual supone una gran disminución de la disipación térmica y, por lo tanto, es un factor más que permite elevar la densidad de integración.

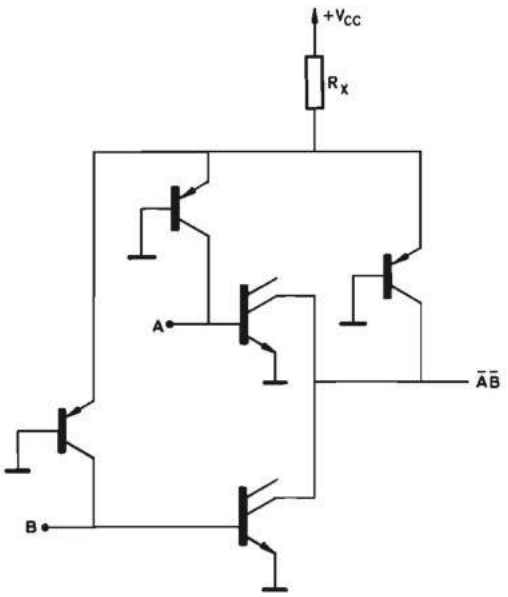


FIGURA 5.59.—**Lógica de inyección integrada (Integrated Injection Logic): Puerta NO-O (NOR) en lógica positiva.**

El desarrollo de las tecnologías de transistores MOS ha hecho que desplazasen a la I^2L pero, no obstante, se sigue utilizando en convertidores analógico-digitales y digital-analógicos, microprocesadores, etc.

5.4.4.4 Circuitos integrados monolíticos digitales con transistores MOS de silicio

5.4.4.4.1 Introducción. La necesidad de realizar circuitos integrados digitales, cada vez más complejos, a fin de aumentar la fiabilidad y la inmunidad al ruido y reducir el tamaño de los sistemas digitales, motivó la utilización de los transistores de efecto de campo MOS.

El menor tamaño de los transistores MOS, la ausencia de difusiones aisladoras y el menor número de difusiones necesario en los circuitos MOS, ha permitido obtener densidades de cinco a seis veces superiores a las de los circuitos bipolares hasta el momento presente.

En la figura 5.60 se representa la configuración física de un transistor MOS constituido por un cristal que puede ser de material *N* o *P* y en el cual se realizan dos difusiones de impurezas de tipo contrario. A ambas difusiones se las denomina drenador (drain) y surtidor (source) y sobre la zona existente entre ellas llamada canal (channel) se deposita una capa de óxido y sobre ésta una capa metálica, que constituye la puerta o graduador (gate).

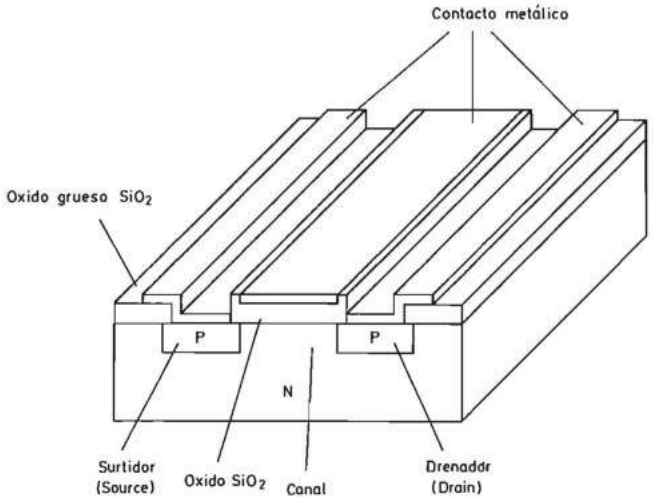


FIGURA 5.60.—**Transistor MOS de canal *P*.**

Solamente cuando el canal tiene una mayoría de portadores iguales a los de las difusiones se puede producir una corriente al aplicar una tensión entre éstas.

Existen dos tipos de transistores MOS que difieren en el tipo de cristal *N* o *P*. Cuando el cristal es del tipo *N*, la corriente entre el surtidor y el drenador es posible cuando el canal se convierte en tipo *P* y, por tanto, este transistor se llama de canal *P*. Por la misma razón cuando el cristal es de tipo *P*, el transistor se denomina de canal *N*. El transistor representado en la figura 5.60 es de canal *P*.

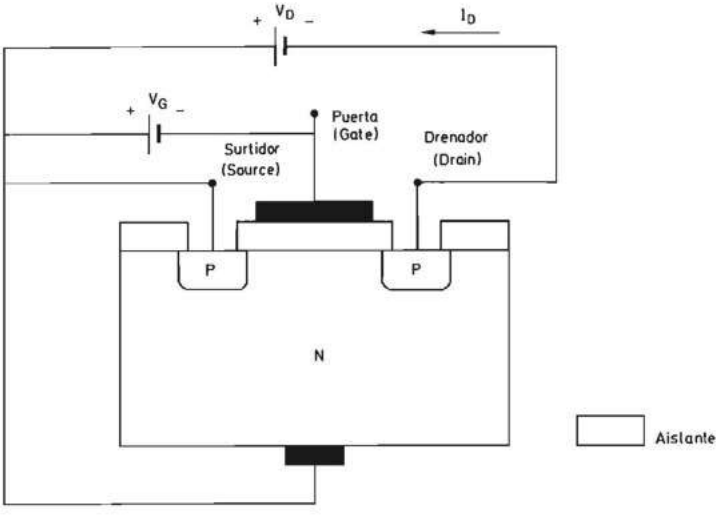


FIGURA 5.61.—**Transistor MOS de canal *P* polarizado.**

Las condiciones de conducción o no conducción del transistor se logran aplicando la tensión correcta a la puerta con respecto al surtidor y al cristal, tal como se indica en la figura 5.61 en la que se representa un transistor MOS de canal *P*.

La zona de cristal situada entre el surtidor (source) y el drenador (drain) puede estar muy impurificada o enriquecida (enhancement) en los portadores de carga del substrato o, por el contrario, poco impurificada o empobrecida (depletion) en dichos portadores, lo cual equivale a estar enriquecida en portadores de signo contrario. La combinación de esta característica con ambos canales *P* y *N* da lugar a cuatro transistores MOS diferentes. En la figura 5.62 se representan las características de intensidad de drenador I_D con respecto a la tensión de graduador (gate) V_G para una tensión entre drenador y surtidor V_D constante, de los cuatro tipos de transistores MOS.

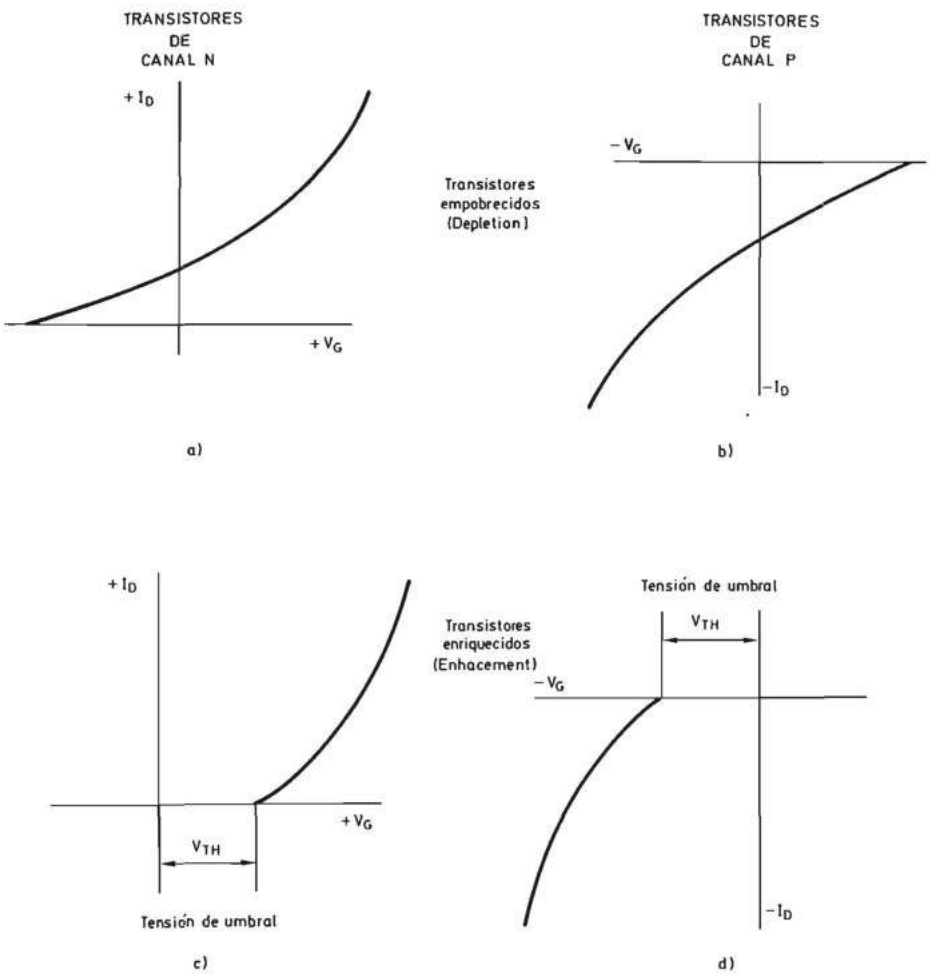


FIGURA 5.62.—**Curvas I_D- V_G de los cuatro tipos de transistores MOS.**

A continuación se analizan las curvas de los transistores de canal N; las de los transistores de tipo P son idénticas sin más que invertir el sentido de la corriente de drenador y la polaridad de la tensión de graduador. En ambos transistores de canal N la intensidad de drenador I_D se eleva al hacerlo la tensión de graduador porque aumenta la anchura de la zona N del canal al ser atraídos los electrones hacia él. En el tipo de canal N poco impurificado o empobrecido (depletion) la zona del canal contiene exceso de electrones en ausencia de tensión de graduador y, por tanto, en estas condiciones existe una cierta corriente de drenador I_D (fig. 5.62*a*). Para anular dicha corriente es necesario polarizar negativamente el graduador con respecto al surtidor para repeler los electrones existentes en la zona del canal. Por el contrario, en el tipo de canal N muy impurificado o enriquecido (enhancement) la zona del canal es de tipo P en ausencia de tensión de puerta y la corriente de colector I_D es nula en estas condiciones. Para que se produzca una corriente de drenador es necesario elevar la tensión de graduador por encima de un cierto valor, que recibe el nombre de tensión umbral (V_{TH}), a partir de la cual la zona del canal se hace de tipo N.

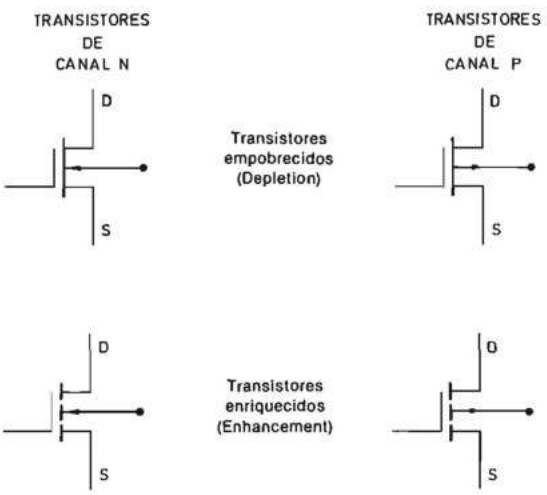


FIGURA 5.63.—**Símbolos utilizados para representar los transistores MOS.**

El rápido desarrollo de los transistores MOS ha hecho que existan diferentes símbolos para representarlos. En la figura 5.63 se indican los primeros símbolos utilizados para los cuatro tipos de transistores MOS. La puerta se representa mediante dos líneas en ángulo recto cuyo vértice está situado más cercano al surtidor que al drenador del transistor. El tipo de canal P se indica mediante una flecha que sale del transistor y el de tipo N por una flecha en sentido contrario. En los transistores enriquecidos en que no hay conducción cuando no se polariza la puer-

ta, se representa discontinua la línea que une el surtidor y el drenador. En los transistores empobrecidos dicha línea se representa continua porque hay conducción cuando la tensión de puerta es nula.

Para simplificar el símbolo se utiliza también la notación representada en la figura 5.64, en la que la flecha que indica si el transistor es de canal N o P se coloca en la conexión de surtidor.

Pero, dado que la flecha se coloca en el surtidor, se puede representar la conexión de la puerta mediante una T en lugar de un ángulo recto, tal como se representa en la figura 5.65.

FIGURA 5.64.—**Símbolos de los transistores MOS empobrecidos obtenidos a partir de los de la figura 5.63: *a*) canal *N*; *b*) canal *P*.**

FIGURA 5.65.—**Símbolos de los transistores MOS enriquecidos con la puerta representada en *T*: *a*) canal *N*; *b*) canal *P*.**

En la figura 5.66 se representan otros símbolos alternativos que se diferencian de los anteriores en que en los transistores enriquecidos se representa continua la línea que une el surtidor y el drenador con un espesor igual a las demás líneas, mientras que en los empobrecidos dicho espesor es mucho mayor.

Finalmente, en la figura 5.67 se representan dos símbolos simplificados utilizados con mucha frecuencia. En ellos se suprimen las flechas y el canal P se indica incluyendo el símbolo de inversión. En las restantes figuras se representan los transistores MOS mediante los símbolos simplificados de la figura 5.67 para los transistores enriquecidos de canal N y canal P. Los transistores empobrecidos se representan mediante un trazo grueso entre el surtidor y el drenador.

En la figura 5.68 se representa la curva característica de corriente de drenador I_D en función de la tensión entre el drenador y el surtidor V_D para diferentes va-

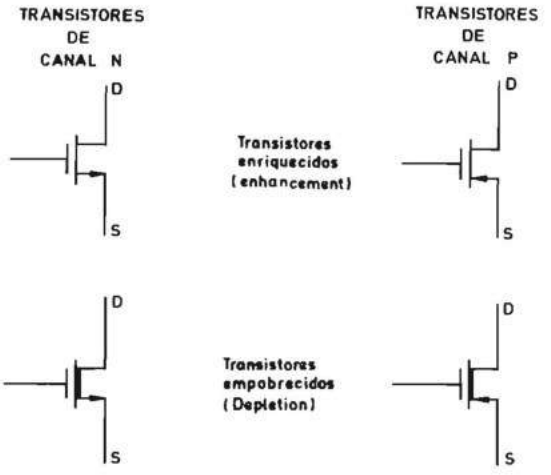


FIGURA 5.66.—**Símbolos alternativos de los transistores MOS.**

FIGURA 5.67.—**Símbolos simplificados de los transistores MOS: *a*) canal *N*; *b*) canal *P*.**

lores de la tensión de graduador de un transistor-canal P enriquecido. El análisis de la figura 5.69 que indica la forma del canal para diferentes valores de la tensión V_D, manteniendo constante la tensión de graduador V_G, permite comprender el porqué de la forma de la característica I_D-V_D de la figura 5.68. El transistor analizado es de canal P, pero el estudio es aplicable al de canal N sin más que invertir la polaridad de V_G y V_D y el sentido de I_D.

Si $V_D = 0$ y el valor negativo de V_G es superior a la tensión de umbral V_{TH}, se forma un canal de anchura constante (fig. 5.69*a*) y una zona poco impurificada. Si, sin variar V_G, se comienza a elevar V_D en sentido negativo, el canal toma la forma de la figura 5.69*b* y la intensidad crece; dicha forma es debida a que la caída de la tensión en el canal se opone a la acción del campo producido por la tensión V_G aplicada al graduador. Si se continúa elevando el valor de V_D en sentido negativo cuando $V_D = V_G - V_{TH}$, el canal llega a anularse en la proximidad del drenador tal como se observa en la figura 5.69*c*; en esas condiciones se pasa de la región de no saturación a la de saturación y, aunque continúe elevándose el va-

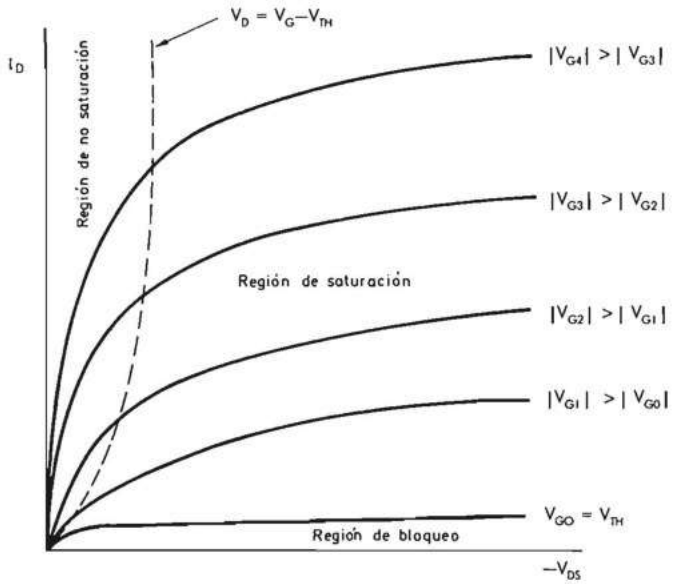


FIGURA 5.68.—**Curva característica I_D - V_{DS} para diferentes valores de la tensión graduador (puerta)-surtidor de un transistor MOS canal *P* enriquecido.**

lor de V_D, la corriente permanece casi constante y el canal se reduce (5.69*d*). La zona en la que el canal se anula es muy pequeña comparada con la distancia total entre las dos difusiones tipo *P* y la corriente I_D circula a través de la zona poco impurificada.

En la figura 5.68 se pueden distinguir tres regiones bien diferenciadas:

a) La región de bloqueo, en la cual el transistor no conduce, es la comprendida entre el eje de abscisas y la curva de tensión de graduador V_G igual a la tensión de umbral V_{TH}.

b) La región de no saturación en la que $V_D < V_G - V_{TH}$ es la que está comprendida entre el eje de ordenadas y la curva que une todos los puntos en que la tensión V_D iguala a la diferencia entre la tensión de puerta y la tensión de umbral.

c) La región de saturación es la comprendida entre las dos curvas antes citadas y en ella la corriente de drenador I_D es prácticamente independiente de la tensión V_D.

El transistor MOS puede ser utilizado en conmutación haciéndolo trabajar entre dos puntos diferenciados de su curva característica I_D-V_D de una forma similar a la del transistor bipolar.

El montaje más elemental utilizable es el de la figura 5.70, en la que se representa un transistor MOS canal *P* enriquecido cuya carga es una resistencia R_L. Haciendo variar la tensión de la puerta (graduador) del transistor entre dos valores $-V_{G1}$ y $-V_{G2}$, el punto de trabajo del transistor cambia de *A* a *B* y la tensión de

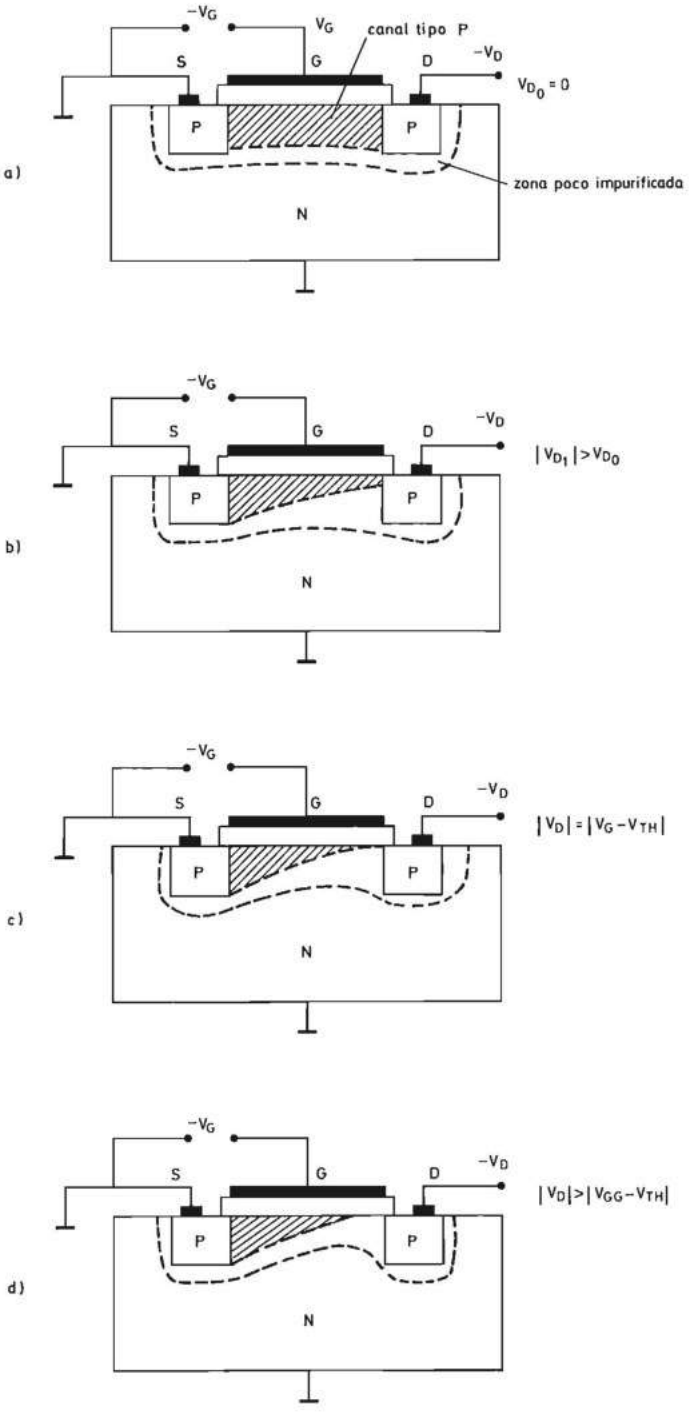


FIGURA 5.69.—**Forma del canal de un transistor MOS tipo *P* para diferentes valores de V_{DS}.**

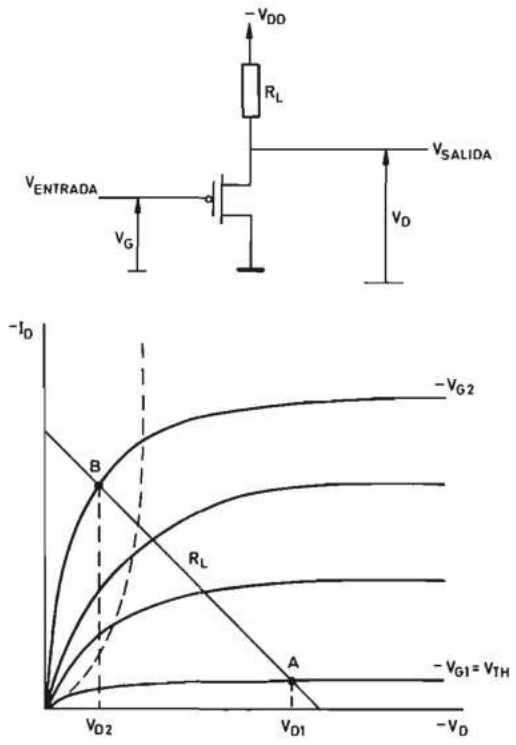


FIGURA 5.70.—**Circuito elemental de un inversor realizado con un transistor MOS de canal *P*.**

salida lo hace entre los niveles V_{D1} y V_{D2}. Asignando a las dos tensiones de entrada y salida los dos valores cero y uno lógicos se comprueba que el circuito de la figura 5.70 constituye un inversor, es decir, si hay un uno lógico a su entrada, su salida es un cero lógico y viceversa. En los circuitos que se estudian a continuación se utilizan transistores canal *N* enriquecidos (enhancement) cuya utilización en la realización de circuitos integrados digitales MOS se ha desarrollado extensamente.

El inversor de la figura 5.70 presenta el inconveniente de que la resistencia R_L ocupa una superficie muy grande. Por ello es preferible utilizar el circuito de la figura 5.71 que se obtiene a partir del de la figura 5.70 sustituyendo la resistencia por otro transistor MOS que actúa como carga. Este transistor MOS que actúa como carga, está diseñado para tener una impedancia unas 20 veces superior a la del transistor que actúa como conmutador a fin de que la tensión de salida sea prácticamente nula cuando éste conduzca.

En el circuito de la figura 5.71 se utiliza como carga un transistor *T*2 enriquecido y por ello es necesario polarizar su puerta con una tensión V_{GG} mayor que V_{DD} para garantizar que *T*2 se encuentra en la región de no saturación cuando *T*1 esté cortado. Se logra así que la caída de tensión en *T*2 sea prácticamente nula en esa situación. Pero la necesidad de utilizar dos tensiones diferentes representa un

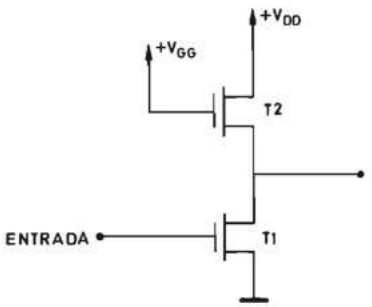


FIGURA 5.71.—**Inversor realizado con un transistor MOS de canal *N* enriquecido, en el que la carga es otro transistor MOS de canal *N* enriquecido.**

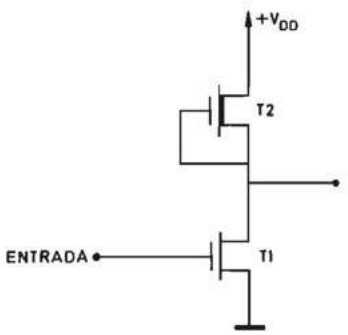


FIGURA 5.72.—**Inversor realizado con un transistor MOS de canal *N* enriquecido, en el que la carga es un transistor MOS de canal *N* empobrecido.**

grave inconveniente, que se supera mediante la utilización como carga de un transistor empobrecido con la puerta unida al surtidor, tal como se indica en la figura 5.72.

Se tiene de esta forma una tensión de polarización fija para *T2* y se garantiza su comportamiento como una resistencia de valor constante.

Mediante la adecuada combinación de transistores MOS se obtienen diversas tecnologías digitales que se estudian en el apartado siguiente.

5.4.4.4.2 Tecnologías digitales MOS

5.4.4.4.2.1 Circuitos integrados MOS estáticos. El conmutador elemental estudiado en el apartado anterior se comporta de una forma similar a uno bipolar y recibe el apelativo de estático en contraposición a los dinámicos que se estudian en el apartado siguiente.

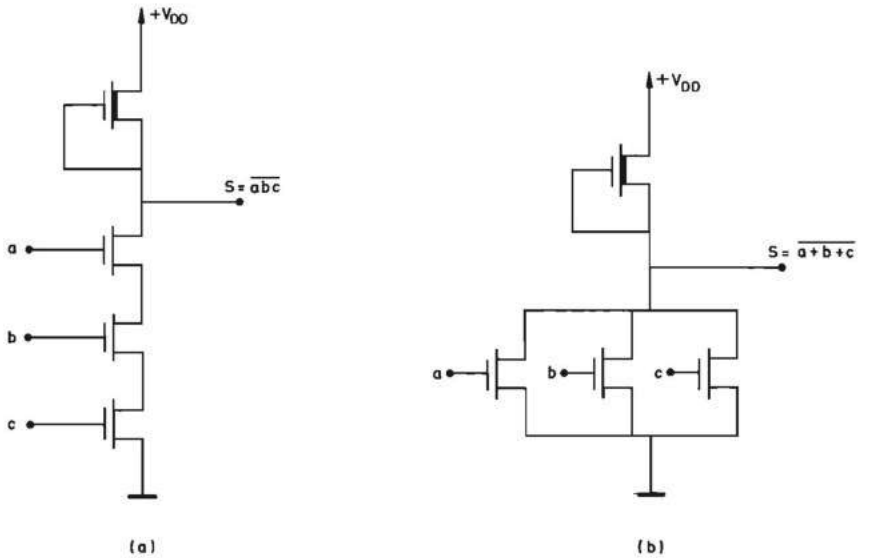


FIGURA 5.73.—**Circuitos MOS estáticos: *a*) puerta NO-Y (NAND) en lógica positiva; *b*) puerta NO-O (NOR) en lógica positiva.**

Combinando diversos circuitos conmutadores se pueden realizar puertas NO-Y y NO-O en lógica positiva, que se representa en la figura 5.73. Se invita al lector a comprobar su funcionamiento.

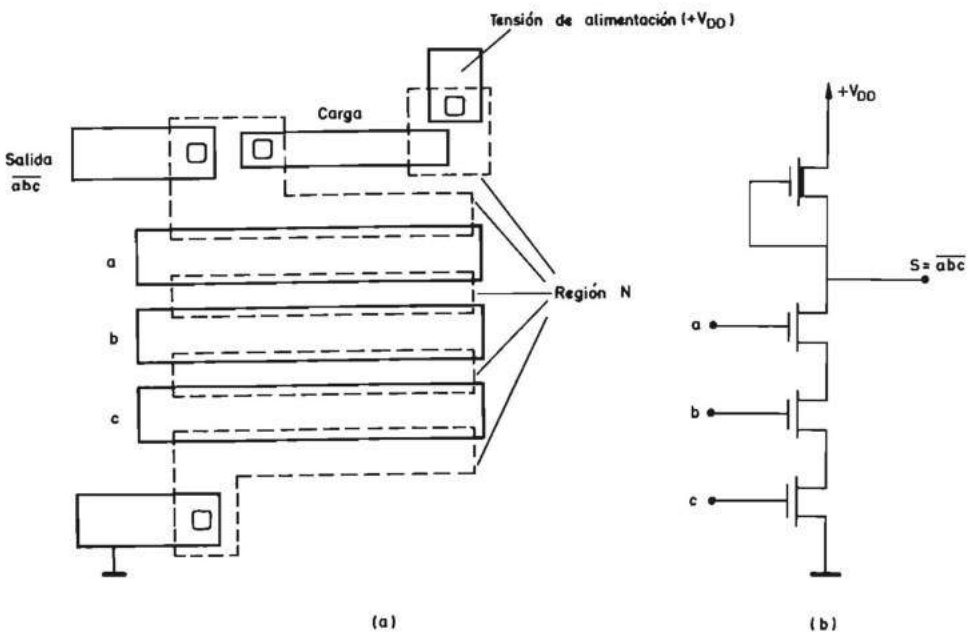


FIGURA 5.74.—**Puerta NO-Y (NAND) en lógica positiva: *a*) realización física; *b*) esquema.**

En la figura 5.74*a* y *b* se representa la realización física de una puerta NO-Y en lógica positiva, en la cual se observa la mayor longitud del transistor que actúa como carga con respecto a los conmutadores.

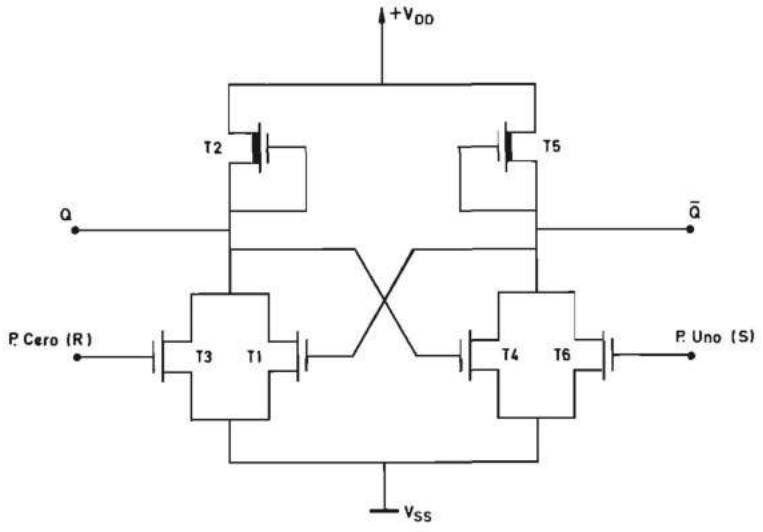


FIGURA 5.75.—**Biestable *R-S* realizado con puertas NO-O (NOR) de MOS estático.**

Mediante circuitos MOS estáticos se realizan biestables *R-S* cuyo esquema se representa en la figura 5.75, que constituyen el bloque básico de las memorias de acceso aleatorio activas. Los transistores *T*1, *T*2 y *T*3 constituyen una puerta NO-Y (NAND) en lógica positiva que se interconecta con la otra puerta NO-Y formada por *T*4, *T*5 y *T*6. El lector puede analizar el funcionamiento.

5.4.4.4.2.2 Circuitos integrados MOS dinámicos. En los circuitos MOS estudiados en el apartado anterior, la puerta de los transistores MOS utilizados como carga se conecta a una tensión fija y reciben el nombre de estáticos. Para reducir la potencia consumida se desarrollaron los circuitos MOS dinámicos que reciben dicho nombre porque la puerta de los transistores que actúan como carga se conecta a una señal cuadrada que conmuta entre dos niveles de tensión que hacen variar la impedancia de la carga entre un valor muy elevado de varios megohmios y el valor nominal (20 veces superior al del transistor conmutador en estado de conducción). Para que no exista pérdida de información y la salida permanezca constante e igual al inverso de la señal de entrada cuando ésta se polariza al nivel de masa, es necesario que la tensión correspondiente se almacene en un condensador. El graduador o puerta de un transistor MOS presenta precisamente la propiedad de ser un condensador casi perfecto con muy pocas pérdidas y por ello, si se conecta a

la salida de un inversor dinámico la puerta de un transistor MOS, no existe pérdida de información si la frecuencia de la onda cuadrada aplicada a la puerta de la carga es suficientemente elevada.

En la figura 5.76 se representa el esquema de un inversor dinámico. El condensador indicado a la salida representa la capacidad puerta-surtidor del transistor MOS que se conecte a ella.

En la figura 5.77 se representa el esquema de las puertas NO-Y y NO-O dinámicas realizadas con transistores de tipo *N* enriquecidos (enhancement).

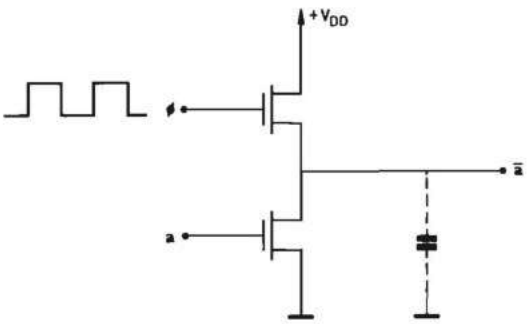


FIGURA 5.76.—**Esquema de un inversor MOS dinámico.**

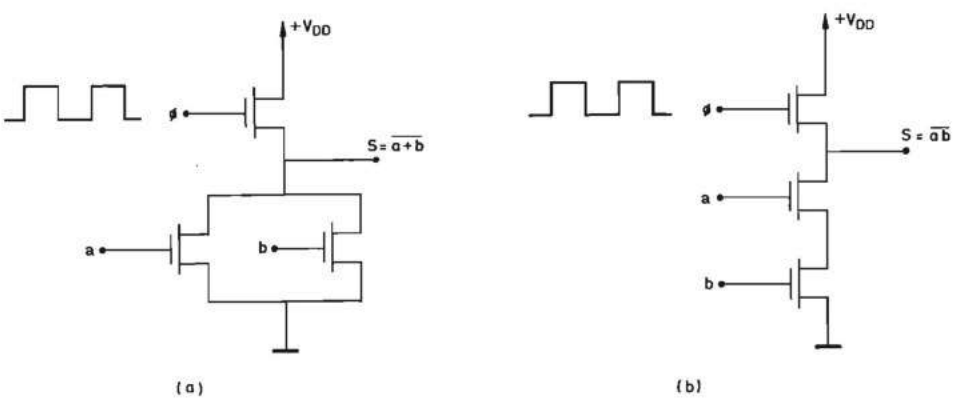


FIGURA 5.77.—**Puertas lógicas en MOS dinámico: *a*) puertas NO-O (NOR); *b*) puertas NO-Y (NAND).**

La ventaja principal de los circuitos de MOS dinámicos es que la potencia consumida cuando la salida está a nivel de masa es igual a la del MOS estático multiplicada por la relación entre la parte del período en que el transistor conduce y el período total.

5.4.4.4.2.3 Circuitos integrados CMOS. Los circuitos estudiados en los dos apartados anteriores utilizan transistores de canal *N* solamente. La combinación de transistores de canal *P* y de canal *N* en un mismo montaje ha dado lugar a una nueva tecnología MOS denominada CMOS (Complementary MOS).

En la figura 5.78 se representa el circuito de un inversor CMOS cuyas principales características se analizan seguidamente. Este circuito está formado por dos transistores, uno de canal *P* y el otro de canal *N*, muy impurificados (enhancement), cuyos drenadores y graduadores se han conectado entre sí. La entrada del circuito es la conexión común de ambos graduadores y la salida la unión común de ambos drenadores. El surtidor del transistor de canal *P* se polariza positivamente con respecto al surtidor del transistor de canal *N*.

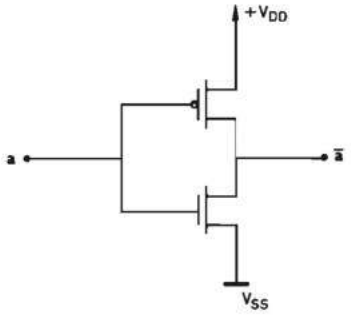


FIGURA 5.78.—**Tecnología CMOS: Circuito de un inversor.**

La tensión de los graduadores puede ser igual a cero o $+V_{DD}$ V. En el primer caso el graduador del transistor de canal *P* se polariza negativamente con respecto a su surtidor y, por tanto, se forma el canal, y la impedancia entre el surtidor y el drenador es muy baja; por el contrario, la tensión del graduador del transistor de tipo *N* es igual a la de su surtidor y por ser muy impurificado presenta una impedancia surtidor-drenador elevada. En estas condiciones la tensión de salida es aproximadamente igual a $+V_{DD}$. En el segundo caso, es decir, cuando la tensión de los graduadores es igual a $+V_{DD}$, se invierten los estados de ambos transistores. En efecto, el graduador del transistor de canal *N* está polarizado positivamente con respecto a su surtidor y se forma el canal que permite la conducción entre el surtidor y el drenador, por el contrario, el transistor de canal *P* no conduce porque el graduador está al mismo nivel de tensión que su surtidor. Por tanto, en este caso la tensión de salida es aproximadamente igual a cero voltios. Se observa que el conjunto de ambos transistores se comporta como un inversor.

La principal característica de este circuito es que no conducen ambos transistores simultáneamente, por lo que la potencia consumida es siempre muy pequeña pues es igual a la tensión de alimentación $+V_{DD}$ por la corriente de pérdidas de un transistor MOS (del orden de décimas de microamperio).

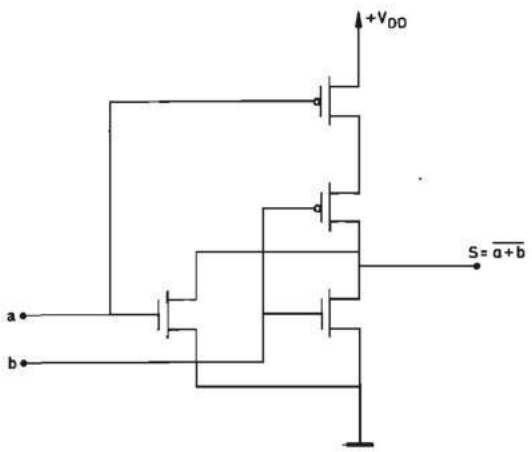


FIGURA 5.79.—**Tecnología CMOS: Esquema de una puerta NO-O (NOR) en lógica positiva.**

La tecnología CMOS permite realizar fácilmente puertas NO-Y y NO-O. En la figura 5.79 se representa el esquema de una puerta NO-O en lógica positiva. Cada entrada se conecta a un transistor de tipo *P* y otro de tipo *N*. Los transistores de tipo *P* se conectan en serie y los de tipo *N* en paralelo. El lector puede comprobar que los dos transistores de tipo *P* sólo conducen simultáneamente cuando ambas entradas están polarizadas al nivel de masa y, por tanto, sólo en dicha situación la tensión de salida es igual a V_{DD}.

En la figura 5.80 se representa el esquema de una puerta NO-Y en lógica positiva. Cada entrada se conecta también a un transistor de tipo *P* y otro de tipo *N*

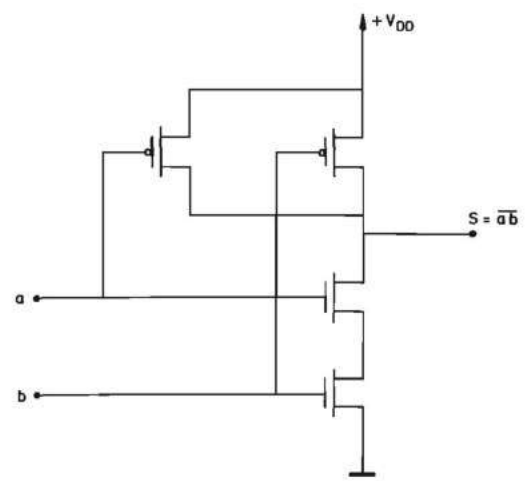


FIGURA 5.80.—**Tecnología CMOS: Esquema de una puerta NO-Y (NAND) en lógica positiva.**

pero, en esta ocasión, los transistores de tipo *P* se conectan en paralelo y los de tipo *N* en serie. El lector puede analizar el funcionamiento de esta puerta.

La tecnología CMOS permite también realizar puertas con salidas de tres estados. En la figura 5.81 se representa un inversor con salida de tres estados. Comparando esta figura con la 5.78 se observa que a esta última se le han añadido los transistores *T*3 (canal *P*) y *T*4 (canal *N*) en serie con los transistores *T*1 y *T*2 que constituyen el inversor. Además, el graduador de *T*3 se conecta a la entrada $\overline{EN}$ y el de *T*4 a la salida del inversor formado por *T*5 y *T*6. Por ello, si $\overline{EN}$ se encuentra en nivel uno, *T*3 y *T*4 están cortados y la salida se encuentra en tercer estado. Si, por el contrario, $\overline{EN}$ está en nivel cero conducen *T*3 y *T*4 y a la salida aparece el inverso de la variable *a*.

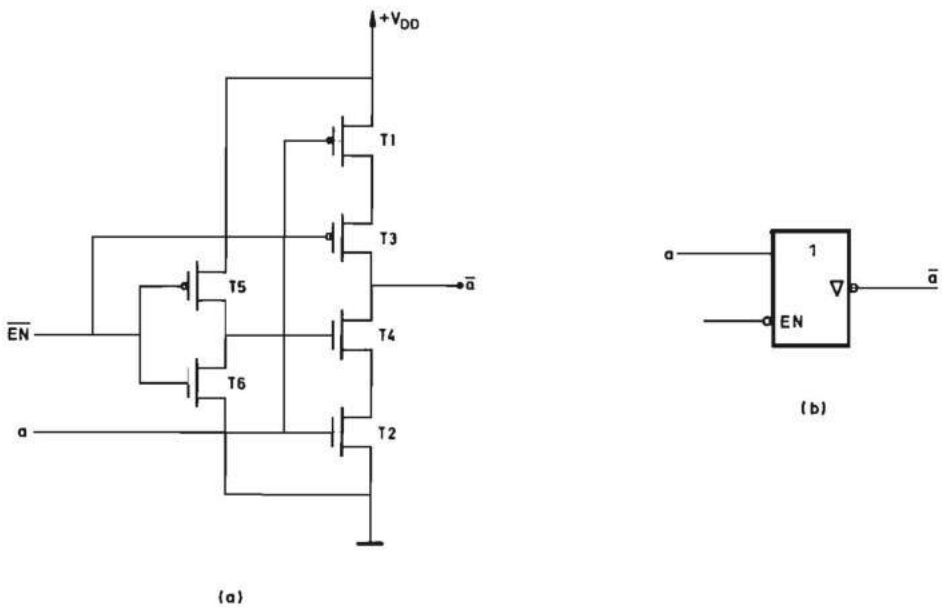


FIGURA 5.81.—**Tecnología CMOS: *a*) esquema del inversor con salida de tres estados; *b*) símbolo lógico normalizado.**

Mediante la interconexión de dos puertas NO-Y (NAND) como las de la figura 5.79 o dos puertas NO-O (NOR) como las de la figura 5.80, se puede realizar un biestable *R-S*. En la figura 5.82 se representa el esquema de un biestable *R-S* realizado con dos puertas NO-O (NOR).

Pero la tecnología CMOS presenta además la característica de permitir la realización de puertas de transmisión que constituyen interruptores electrónicos cuya impedancia varía entre algunos ohmios y varios megohmios en función del estado lógico de una variable binaria.

Las puertas de transmisión son difícilmente realizables en tecnología bipolar debido a la falta de aislamiento entre la base y el colector de los transistores bipo-

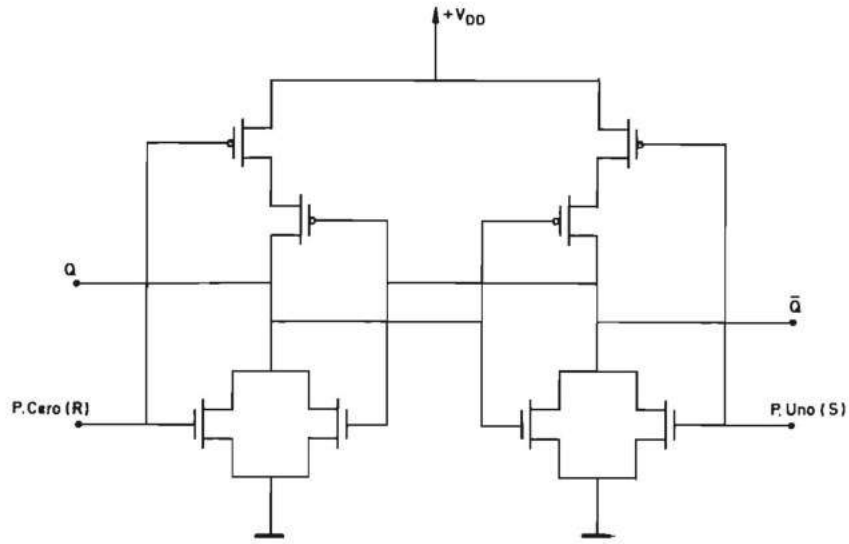


FIGURA 5.82.—**Tecnología CMOS: Biestable *R-S* realizado con dos puertas NO-O (NOR).**

lares. Pero también en MOS estático con transistores de un solo tipo de canal (como por ejemplo el *N*) presentan el inconveniente de que el valor de la resistencia del transistor depende de la tensión puerta-surtidor tal como se indica en la figura 5.83. Si la tensión V_{GS} es nula o negativa, la resistencia R es muy elevada. Por ello, si V_e varía entre $+V$ y 0 V con respecto al común, el transistor está cortado si la tensión de control V_c es cero.

Pero en el caso en que $V_c = +V$, la resistencia R depende del valor de V_e. En efecto V_e ha de ser menor que $V - V_{UMBRAL}$ en un cierto valor para que R tenga un valor reducido.

El problema descrito se solventa mediante la colocación de dos transistores en paralelo, uno de canal *P* y otro de canal *N* con sus puertas controladas por señales

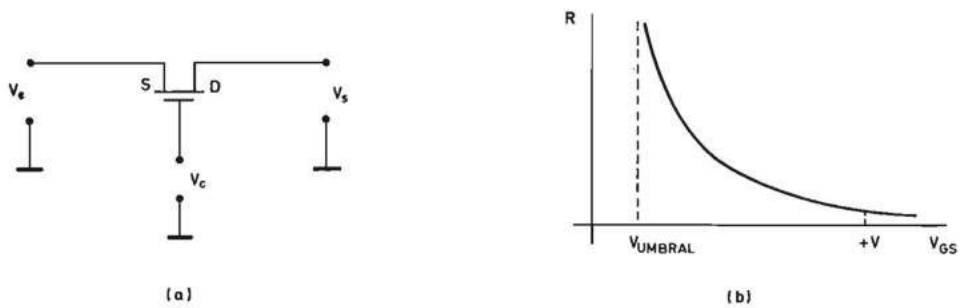


FIGURA 5.83.—**Puerta de transmisión realizada con un transistor MOS canal *N* enriquecido: *a*) esquema; *b*) curva R-V_{GS}.**

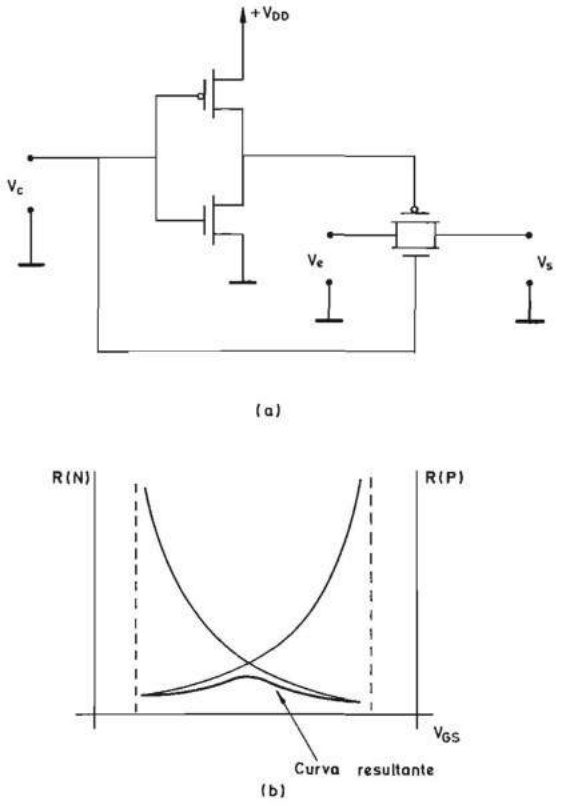


FIGURA 5.84.—**Puerta de transmisión en tecnología CMOS: *a*) esquema; *b*) curva R-V_{GS}.**

binarias inversas, que constituye un circuito en tecnología CMOS que se representa en la figura 5.84*a*. En efecto, si la tensión de control V_c es cero, la puerta del transistor de canal P está a $+V$ y la del transistor de canal N está a cero voltios y ambos están cortados. Si, por el contrario, V_c es igual a $+V$, la puerta del transistor de canal P está a cero voltios y la del transistor de canal N está a $+V$; si en esta situación V_e es igual a cero, es el transistor N el que posee una impedancia baja y el P la tiena alta. Al ir elevándose V_e la impedancia del transistor N sube y la del P baja hasta alcanzar la situación inversa cuando $V_e = +V$. La impedancia resultante del paralelo de ambos transistores es, por lo tanto, prácticamente constante tal como se representa en la figura 5.84*b*.

Las puertas de transmisión permiten realizar multiplexores/demultiplexores por el método descrito en el apartado 3.7.3.3 del capítulo 3. En la figura 5.85 se representa un multiplexor/demultiplexor de dos canales realizado con sendas puertas de transmisión, que utiliza dos puertas de transmisión controladas con una única variable de selección que se conecta de forma directa al transistor P de una puerta y al N de la otra, y de forma inversa a las otras dos. Mediante puertas de transmi-

sión y un decodificador se puede realizar un multiplexor/demultiplexor de cualquier número de canales que puede trabajar tanto con señales digitales como analógicas.

La utilización de multiplexores con puertas de transmisión y otros elementos lógicos simplifica la realización de numerosos circuitos lógicos. Por ejemplo, la utilización de un multiplexor de dos canales y sendos inversores da lugar a un biestable *D* activado por niveles o cerrojo (latch) cuyo esquema lógico se representa en la figura 5.86. El lector puede convertir este esquema en el correspondiente en tecnología CMOS con puertas de transmisión y comprobar su gran sencillez.

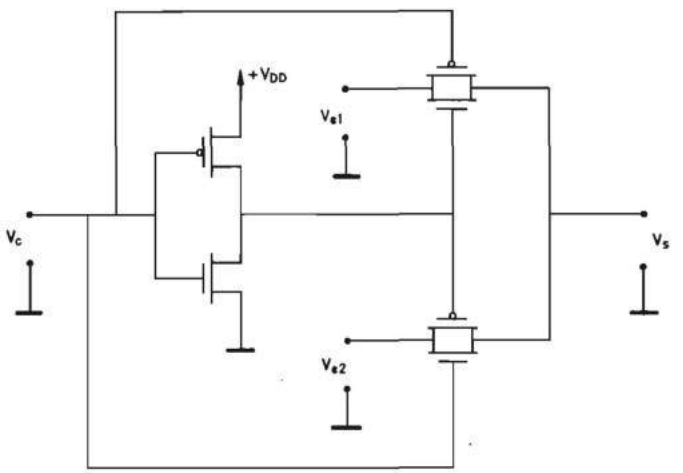


FIGURA 5.85.—**Tecnología CMOS: Esquema de un multiplexor/demultiplexor de dos canales.**

Como resumen de todo lo expuesto se puede indicar que la tecnología CMOS presenta las siguientes características:

a) Una mayor complejidad en sus circuitos que las tecnologías de MOS estático o dinámico (compárense las figuras 5.75 y 5.82 por ejemplo) que ha hecho que su uso se generalizase al alcanzarse un elevado grado de capacidad de integración por parte de los fabricantes.

b) Una potencia consumida mucho menor que las otras dos, que es prácticamente nula cuando los circuitos permanecen en reposo. Esto hace a esta tecnología idónea para realizar sistemas que mantengan la información en ausencia de la tensión de red, quedando alimentados por baterías.

c) Realización de las puertas lógicas exclusivamente con transistores MOS, característica común a las otras dos tecnologías MOS. Esto permitió alcanzar elevadas densidades de integración.

d) Realización de puertas de transmisión que simplifican la realización de los circuitos secuenciales.

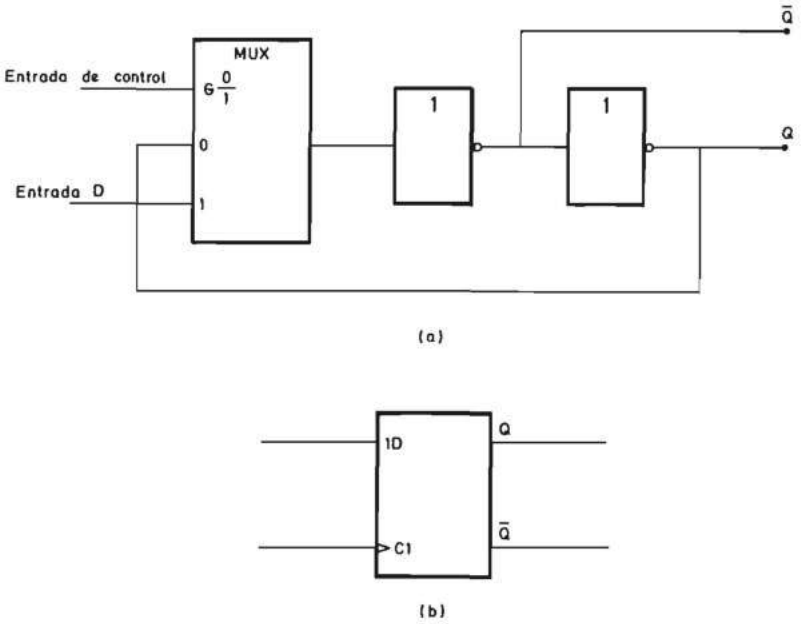


FIGURA 5.86.—**Tecnología CMOS: *a*) esquema lógico de un biestable *D* activado por niveles o cerrojo (latch); *b*) símbolo lógico normalizado.**

e) Elevado margen de ruidos debido a que tanto los transistores *P* como los *N* utilizados son del tipo enriquecido.
f) Elevada cargabilidad de salida (fan-out) debido a que el graduador o puerta de un transistor MOS presenta una impedancia capacitiva prácticamente pura y, por lo tanto, su consumo es nulo cuando la tensión permanece constante.

5.4.4.5 Circuitos integrados digitales con transistores bipolares y MOS (BICMOS)

Las tecnologías MOS, y muy especialmente la CMOS, han sido las que han alcanzado un gran desarrollo debido a su bajo consumo y a la mejora de las técnicas de fabricación que han permitido el diseño de los circuitos VLSI y ULSI. No obstante, con transistores bipolares de las mismas dimensiones se logran circuitos dos veces más rápidos y con mucha mayor cargabilidad de salida (driving capability).

Por todo ello surgió el interés de una tecnología que combinase el bajo consumo del CMOS con la elevada cargabilidad de los transistores bipolares para obtener circuitos lógicos especialmente orientados a la realización de barras de información entre circuitos VLSI y ULSI como microprocesadores, memorias y unidades de acoplamiento de periféricos.

En la figura 5.87 se representa un esquema posible de una puerta Y (AND) en lógica positiva de esta tecnología que recibe el nombre de BICMOS. Este circui-

to tiene a su entrada una puerta NO-Y (NAND) de tecnología CMOS seguida de un transistor MOS canal *N* en divisor de fase unido a un montaje inversor con carga activa realizado con transistores bipolares Schottky. Cuando cualquiera de las dos entradas *a* o *b* se encuentra en nivel cero, se pone a nivel $+V_{CC}$ la puerta del transistor MOS *T*5; esto hace que se sature *T*8 y se corten *T*6 y *T*7 porque la tensión en la base de *T*6 es menor que la de umbral.

Si, por el contrario, *a* y *b* se encuentran en nivel uno ($+V_{CC}$), la puerta de *T*5 está a nivel cero y, por lo tanto, este transistor está cortado. Esto hace que se corte *T*8 y que, por el contrario, conduzcan *T*6 y *T*7.

Para un estudio más profundo de la tecnología BICMOS, el lector puede consultar la bibliografía [TEXA 89].

5.4.4.6 Circuitos integrados digitales de arseniuro de galio (GaAs)

5.4.4.6.1 Introducción. Todos los circuitos integrados digitales estudiados en apartados anteriores utilizan el silicio como componente.

Pero un semiconductor se puede formar mediante una mezcla de elementos del grupo III y del grupo V. Dichos compuestos no se encuentran en la naturaleza, y deben sintetizarse artificialmente.

H. Welker descubrió en los años cincuenta que la naturaleza del enlace químico de estos compuestos permitía obtener una gran movilidad de los electrones. Sin embargo, las características del silicio y especialmente la existencia del óxido de silicio (SiO_2) permitieron el desarrollo de los transistores planar y su integración en

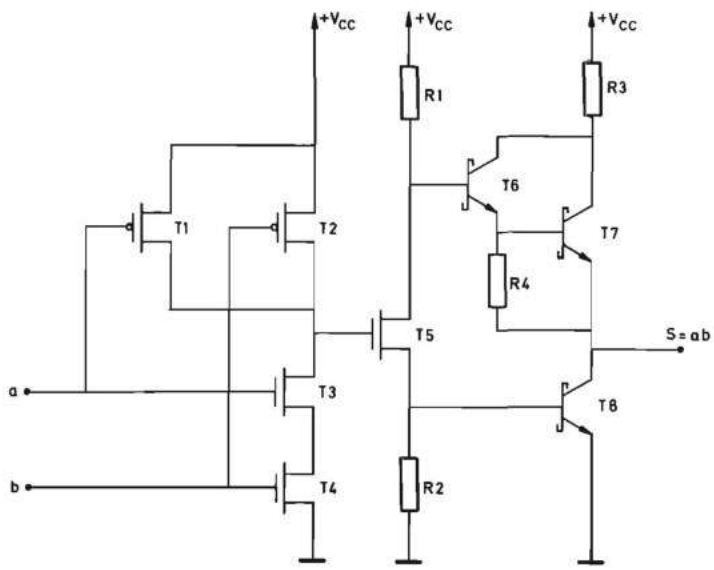


FIGURA 5.87.—**Esquema de una puerta Y (AND) en lógica positiva en tecnología BICMOS.**

circuitos bipolares y posteriormente el de los transistores de efecto de campo de puerta aislada (MOS).

Paulatinamente, la utilización del silicio se generalizó de tal manera que existe poco interés en industrializar otra tecnología en aquellas aplicaciones que pueden resolverse con dispositivos de silicio.

Sin embargo, el silicio no es un semiconductor universal que dé respuesta a todos los problemas que se presentan al tratar de realizar dispositivos en distintas situaciones prácticas. Por ejemplo, las tecnologías bipolares de silicio difícilmente permiten obtener puertas lógicas con un tiempo de propagación inferior al nanosegundo aunque se utilicen técnicas que impidan la saturación de los transistores.

Por ello al alcanzar el silicio los límites tecnológicos prácticos se ha producido el interés de desarrollar dispositivos con semiconductores III-V y, en concreto, con arseniuro de galio (GaAs). Los dispositivos de GaAs son, por lo tanto, complementarios de los de silicio en la fabricación de circuitos integrados de gran velocidad y permiten, debido a la gran movilidad de los electrones en ellos, realizar puertas con un tiempo de propagación no superior a los 100 picosegundos.

5.4.4.6.2 Dispositivos de arseniuro de galio. Los transistores MOSFET han sido ampliamente utilizados en tecnologías de silicio debido a las características estables del óxido de silicio (SiO_2) que permiten su utilización como aislante entre la puerta y el sustrato. Por el contrario, los óxidos de arseniuro de galio presentan grandes dispersiones en sus características que no permiten obtener tensiones de umbral constantes.

Por ello se han realizado grandes esfuerzos, con resultados positivos, para realizar en tecnologías GaAs transistores de efecto de campo de semiconductor y metal [Metal-semiconductor field effect transistor (MESFET) una de cuyas estructuras básicas se representa en la figura 5.88.

Este transistor se diferencia del MOS en la utilización de un sustrato semiaislante formado por una zona de elevada resistividad que permite que la tensión aplicada entre él y la puerta controle por efecto de campo la anchura del canal entre el drenador y el surtidor o fuente.

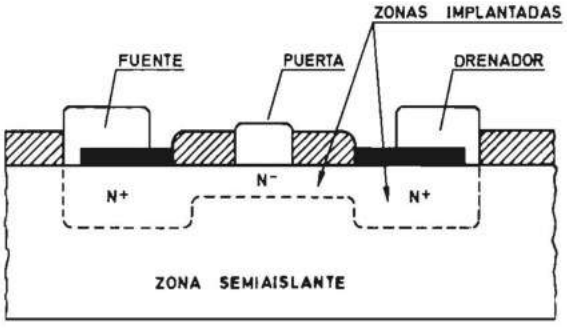


FIGURA 5.88.—**Estructura básica de un transistor MESFET de arseniuro de galio.**

Los transistores MESFET pueden ser, al igual que los MOS, empobrecidos (depletion) D-MESFET y enriquecidos (enhancement) E-MESFET. Los D-MESFET conducen con tensión de puerta nula y los E-MESFET necesitan ser polarizados adecuadamente (positivamente la puerta con respecto al sustrato en transistores de canal *N*).

Los transistores empobrecidos (D-MESFET) de canal *N* presentan la desventaja de necesitar una tensión negativa en la puerta para ser llevados al corte, lo cual implica dos tensiones de alimentación en lugar de una sola. Por el contrario, se han logrado transistores D-MESFET de una gran estabilidad que han permitido la realización de circuitos de gran y muy gran escala de integración.

Los transistores E-MESFET, que como se indicó anteriormente no conducen en ausencia de tensión de control en la puerta, necesitan una única tensión de alimentación y permiten la realización de puertas lógicas de acoplamiento directo [direct-coupled FET logic (DCFL)]. En la figura 5.89*a* se representa una puerta NO-O (NOR) en lógica positiva y en la figura 5.89*b* una puerta inversora [PIQU 84].

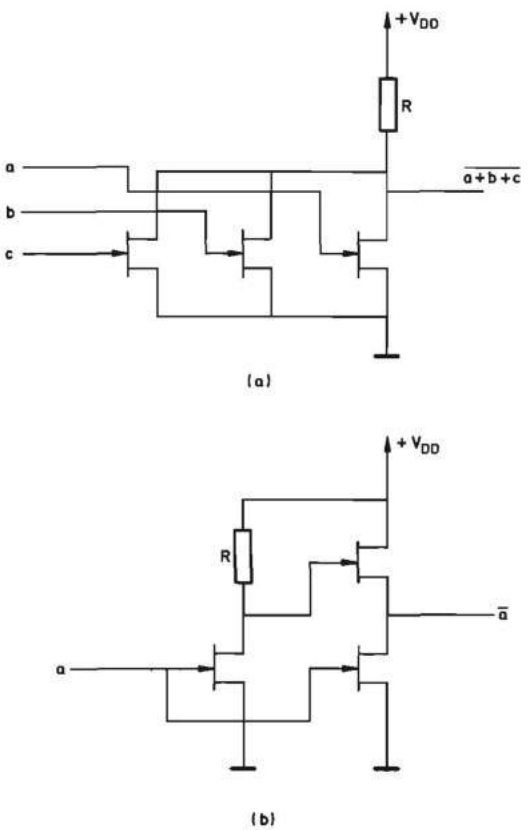


FIGURA 5.89. —**Puertas realizadas con transistores MESFET enriquecidos: *a*) puerta NO-O (NOR); *b*) puerta inversora.**

BIBLIOGRAFIA

[ACTE 95] «Act family FPGA data book», Actel Corporation, EUA, 1995. *http://www.actel.com*
[ALTE 88] «User configurable data book». Altera 1988.
[ALTE 90] «Flex and Max families data book». Altera 1990.
[ALTE 95] «Altera Data Book». Altera Corporation. San José (CA). 1995.
[AMD 89] «Mach devices data book». AMD 1989.
[ALVA 95] Alvarez, J., Metodologia de diseño de controladores lógicos caracterizados por flancos realizados con circuitos digitales configurables. Tesis Doctoral, Departamento de Tecnología Electrónica, Universidad de Vigo, diciembre 1995.
[AT&T 95] «AT&T ORCA™ ATT2CxxA Series FPGAs», AT&T Microelectronics, noviembre 1995.
[BURS 96] Bursky D., «CPLDs and dedicated memory, counters to up performance», Electronic Design, marzo 1996.
[CROS 92] «Crosspoint Data Book», Crosspoint Solutions, Inc., 1992.
[GEIG 90] Randall L, Geigeir Phillip E., Allen Noel R. Strader VLSI Design techniques for analog and digital circuits. McGraw-Hill Publishing Company 1990.
[HORB 86] E. Horbst. Advances in CAD for VLSI. Volume 2. Logic Design and Simulation. North Holland 1986.
[HURS 85] Stanley L. Hurst. Custom-Specific integrated circuits. Marcel Dekker Inc. 1985.
[INTE 90] «Programmable logic handbook». Intel 1990.
[INTE 94] «Programmable Logic», Intel Corporation, 1994.
[LATT 94] «Lattice Handbook», Lattice Semiconductor Corporation, Oregon, EUA, 1994. *http://www.lattice.com*
[LATT 95] «ispLSI and pLSI 1048E High-Density Programmable Logic», Lattice Semiconductor Corporation, Oregon 1995. *http://www.lattice.com*
[LIPO 88] G. J. Lipovski. Single and Multiple-chip microcomputer interfacing. Apartado 5.2.4. Reconocimiento de interrupciones por consulta (polled interrupts). Prentice-Hall Inc. 1988.
[MAND 99] E. Mandado, J. Fariña, E. Soto. Sistemas electrónicos digitales. Tomo II: Procesadores digitales secuenciales. Editorial Marcombo, 1999.
[MILL 88] J. Millman & Arvin Grabel. Microelectronics. The BJT as a switch pp. 110 a 114. McGraw-Hill. 1988.
[PERE 91] Pérez S., «Nuevos métodos sistemáticos de diseño de controladores lógicos». Tesis Doctoral, Departamento de Tecnología Electrónica, Universidad de Vigo, 1991.
[PHIL 90] «Semi-custom programmable logic devices (PLD) data handbook». Philips 1990.
[PIQU 84] J. Piqueras. Circuitos integrados de GaAs. Desafío tecnológico y perspectivas de futuro. Mundo Electrónico, nº 145. 1984.
[SEID 83] Arthur H. Seidman. Integrated circuits applications handbook. John Wiley & Sons. 1983.
[TEXA 89] BICMOS Bus interface logic data book. Texas Instruments 1989.
[TEXA 93] «FPGA Data Manual». Texas Instruments 1993. *http://www.ti.com*
[XILI 94] «The Programmable Logic Data Book», Xilinx, San José (CA), 1994.
[XILI 95] «XC8100 FPGA family», Xilinx, San José (CA), octubre 1995. *http://www.xilinx.com*
[XILI 96] «XC4000 Series FPGAs», Xilinx, San José (CA), septiembre 1996. *http://www.xilinx.com*

Capítulo

6

Sistemas secuenciales

6.1 DEFINICION

En el capítulo 3 se ha realizado el análisis y la síntesis de los sistemas lógicos combinacionales, en los que no se ha tenido en cuenta la variable tiempo porque el estado lógico de la salida de dichos sistemas depende solamente del estado de las entradas en el instante considerado y es independiente de los estados que hayan tenido las entradas anteriormente. Por tanto, los sistemas digitales combinacionales no permiten por sí solos resolver el problema de almacenar el estado de las entradas en un instante y utilizarlo para tomar decisiones posteriormente cuando cambie el estado de aquéllas.

Los sistemas que puedan actuar de la forma anteriormente descrita han de ser capaces de memorizar el estado de las entradas y convertirlo en un estado interno del propio sistema. El valor de la salida en un instante determinado no depende solamente del estado de las entradas en dicho instante sino también del estado interno.

De todo lo dicho se deduce que un sistema secuencial reacciona ante secuencias de estados de entrada de una forma determinada, que depende de su construcción física y, por tanto, es un sistema automático, que recibe el nombre de autómata finito debido a que posee un número finito de estados internos. La denominación de secuencial es debida a que el valor de su salida en un instante determinado no depende solamente de los estados lógicos de las entradas en dicho instante sino de la secuencia de estados de las entradas.

Todo sistema secuencial posee:

a) un conjunto finito de 2^n estados de entrada siendo n el número de variables de entrada $x_1, x_2 \ldots x_n$;

b) un conjunto finito de 2^m estados internos siendo m el número de variables de estado interno $y_1, y_2 \ldots y_m$;

c) un conjunto finito de 2^p estados de salida siendo p el número de variables de salida $z_1, z_2 \ldots z_p$.

Las variables de salida se pueden obtener de dos formas diferentes:

a) Mediante combinación de las variables de entrada y las variables de estado

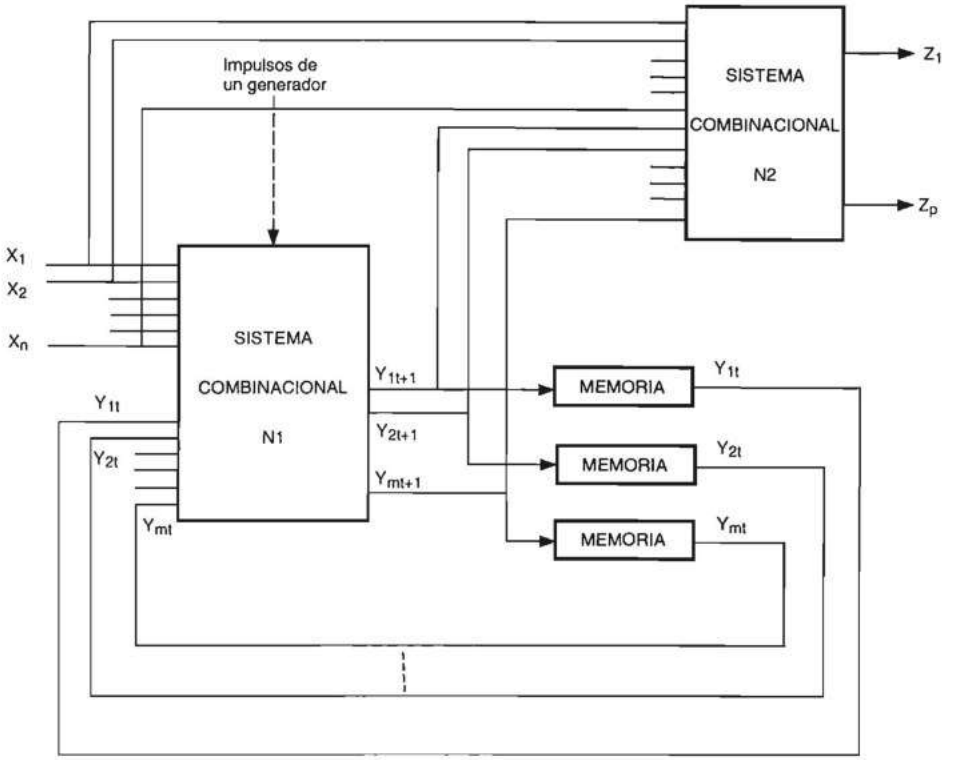


FIGURA 6.1.—**Diagrama de bloques de un autómata de Mealy.**

interno. La estructura de un sistema secuencial de este tipo, que recibe el nombre de autómata de Mealy, corresponde al diagrama de bloques de la figura 6.1.

b) Las variables de salida coinciden con las variables de estado interno o dependen solamente de ellas. La estructura de un sistema secuencial de este tipo, que recibe el nombre de autómata de Moore, corresponde al diagrama de bloques de la figura 6.2.

Se demuestra algebraicamente que para todo autómata de Moore existe un autómata de Mealy equivalente y viceversa.

El autómata más general es el de Mealy representado en la figura 6.1 y que está formado por dos sistemas combinacionales N1 y N2. Se puede considerar que ambos sistemas constituyen una multifunción que genera ambos tipos de variables, las de estado interno y las de salida. Se obtiene asi el diagrama de la figura 6.3 del que los de las figuras 6.1 y 6.2 son casos particulares.

El sistema combinacional recibe en sus entradas las variables externas x que constituyen las variables de entrada del autómata y las variables de estado interno y realimentadas a través de unos elementos de memoria. Según cómo se realicen estas memorias se tienen diferentes tipos de sistemas secuenciales que se describen brevemente a continuación y se estudian con más detalle en apartados sucesivos.

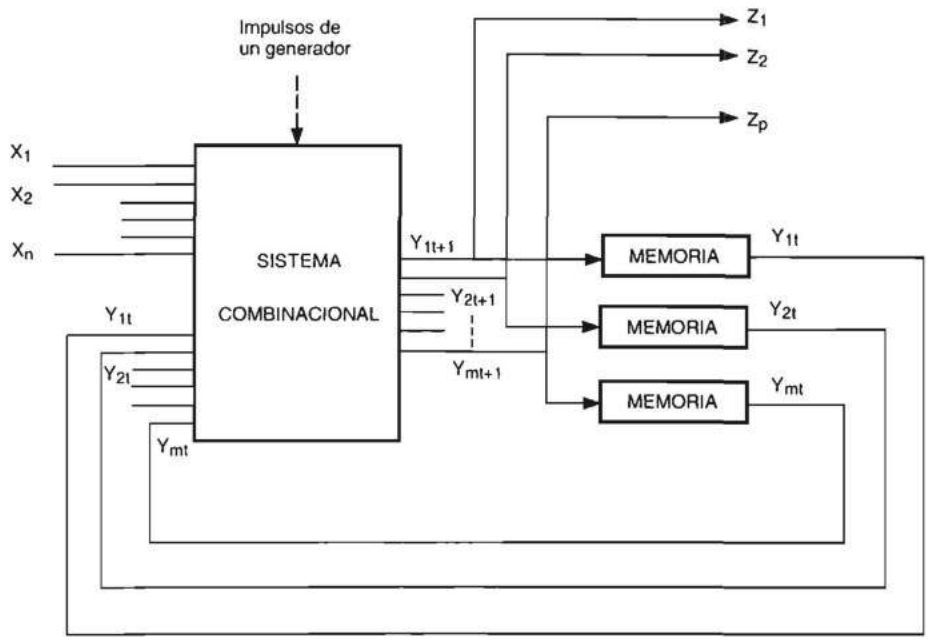


FIGURA 6.2.—**Diagrama de bloques de un autómata de Moore.**

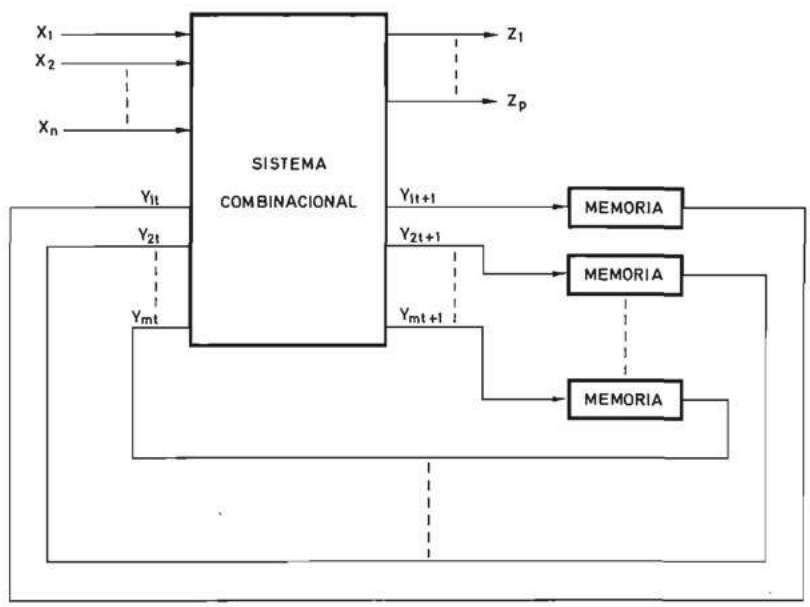


FIGURA 6.3.—**Diagrama de bloques de un autómata.**

Las memorias indicadas en las figuras 6.1, 6.2 y 6.3 pueden estar constituidas de las maneras siguientes:

a) Por el tiempo de propagación de las puertas que forman el sistema combinacional. Parte de las variables de salida de este sistema se realimentan directamente a la entrada tal como se representa en la figura 6.4 y por ello el autómata obtenido se denomina de reacción o realimentación directa. En este sistema las variables de entrada actúan de forma directa sobre el autómata que recibe el nombre genérico de asíncrono. Se trata por lo tanto de un autómata asíncrono a reacción directa.

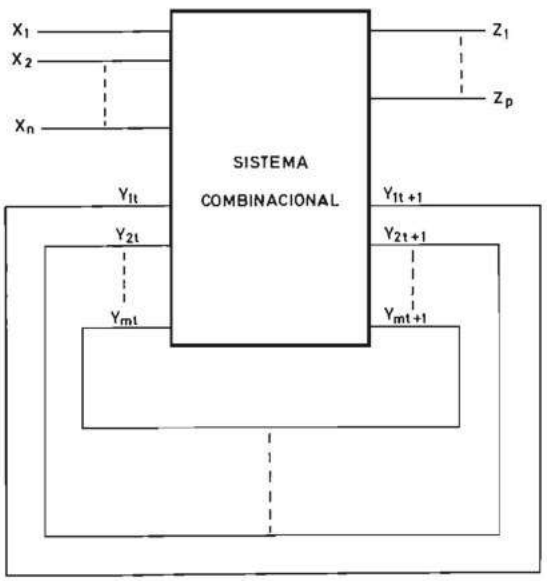


FIGURA 6.4.—**Diagrama de bloques de un autómata asíncrono a reacción directa.**

b) Células secuenciales básicas con una única variable de salida que constituyen un autómata elemental. Estas células pueden ser de diferentes tipos que dan lugar a otras tantas variantes de autómatas.

b-1) Células en las que las variables de entrada actúan de forma directa sobre el estado de las mismas y que constituyen realmente autómatas asíncronos de realimentación directa. Se obtiene así un autómata asíncrono con células elementales cuyo diagrama de bloques se representa en la figura 6.5. Las células utilizadas pueden ser de dos tipos distintos que corresponden a diferentes formas de actuación de las variables de entrada sobre el estado de la célula. Si las variables de entrada actúan mediante la adopción de determinados niveles lógicos, el autómata obtenido se caracteriza por niveles. Si, por el contrario, las variables de entrada actúan mediante los cambios de nivel en sus entradas se obtiene un autómata asíncrono caracterizado por flancos.

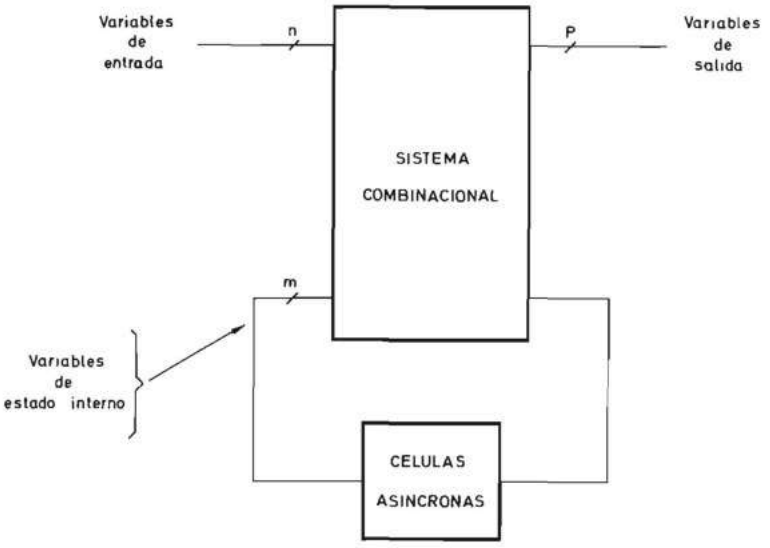


FIGURA 6.5.—**Diagrama de bloques de un autómata asíncrono con realimentación a través de células elementales.**

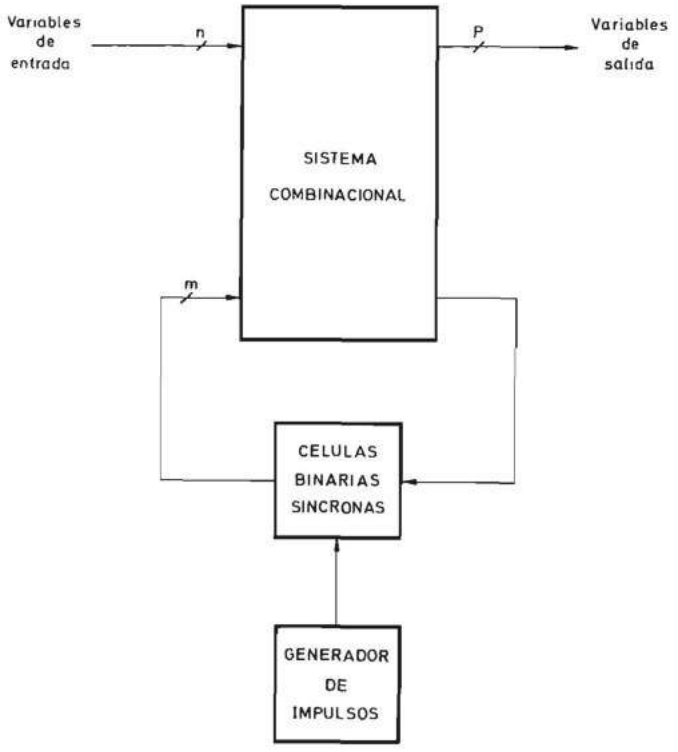


FIGURA 6.6.—**Diagrama de bloques de un autómata síncrono.**

b-2) Células en las que las variables de entrada actúan solamente sobre el estado de las mismas en el instante en que aparece un impulso en una entrada de sincronización. El sistema obtenido se representa en la figura 6.6. Las células que realimentan parte de las variables de salida a la entrada del sistema combinacional están sincronizadas con los impulsos de un generador. Según la forma en que se realiza dicha sincronización, el sistema secuencial puede ser totalmente síncrono o síncrono asincronizado. En el apartado 6.3 se estudian las características de estos sistemas secuenciales.

De acuerdo con lo expuesto, los sistemas secuenciales se clasifican en dos grandes familias constituidas por los asíncronos y los síncronos tal como se indica en la tabla 6.1.

En sucesivos apartados se estudian las diferentes clases de sistemas secuenciales.

SISTEMAS SECUENCIALES	Asíncronos	De realimentación directa
		Realimentados con células asíncronas
	Síncronos	Totalmente síncronos
		Síncronos asincronizados

TABLA 6.1

6.2 SISTEMAS SECUENCIALES ASINCRONOS

6.2.1 Introducción

Los sistemas secuenciales asíncronos son aquellos en los que las variables de entrada actúan de forma directa sobre el sistema. Los cambios de estado adecuados de las variables de entrada producen directamente un cambio de estado interno.

Los sistemas secuenciales asíncronos se pueden caracterizar mediante los estados (niveles) de las variables de entrada o los cambios de estado de estas variables. Esto da lugar a dos métodos de diseño diferentes (tabla 6.2) que se estudian en apartados sucesivos.

SISTEMAS SECUENCIALES ASINCRONOS	Caracterizados por niveles	*a*) Realimentación directa
		b) Realimentación con biestables
	Caracterizados por flancos	Realimentación con células activadas por flancos

TABLA 6.2

6.2.2. Sistemas secuenciales asíncronos caracterizados mediante niveles

Estos sistemas son los más intuitivos y los que se desarrollaron más rápidamente. Se pueden realizar de dos formas diferentes, que se estudian a continuación.

6.2.2.1 Sistemas secuenciales asíncronos de realimentación directa. Tal como su denominación indica, estos sistemas secuenciales están constituidos por un sistema combinacional parte de cuyas entradas están unidas a sus salidas tal como se representa en la figura 6.4.

La memoria está constituida por el tiempo de propagación a través de las puertas del sistema combinacional.

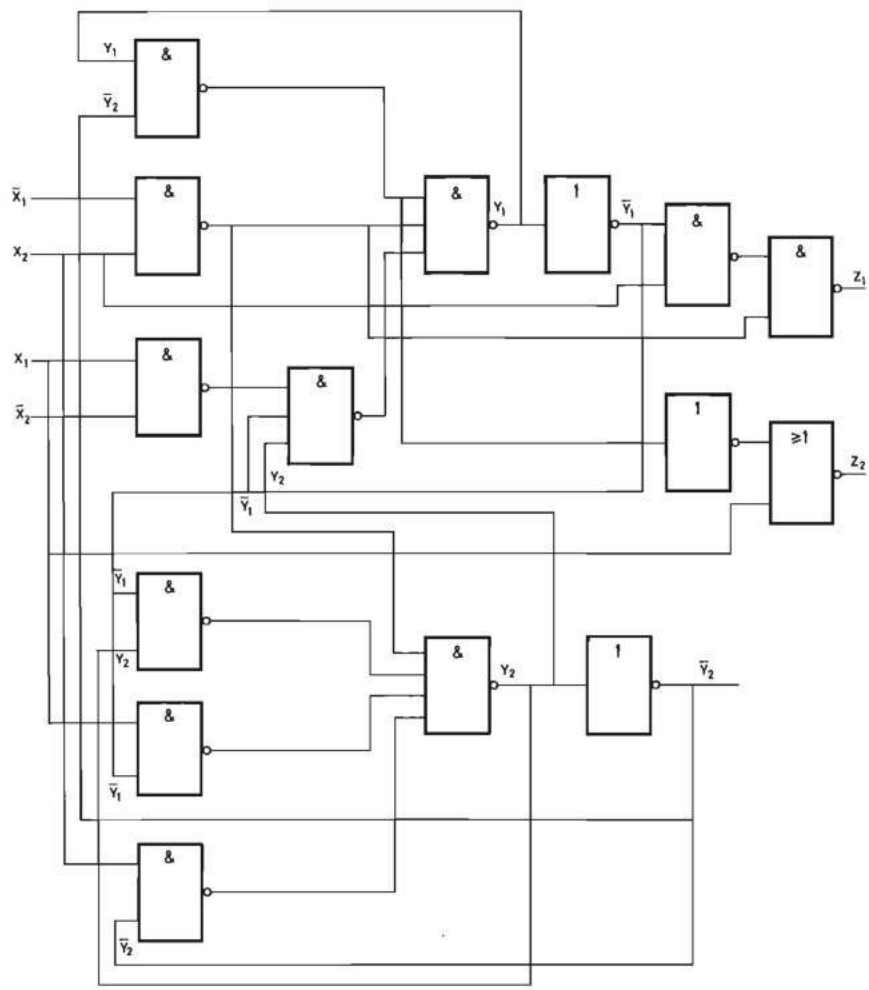


FIGURA 6.7.—**Ejemplo de sistema secuencial asíncrono de realimentación directa.**

Estos sistemas fueron los primeros que se analizaron y su estudio resulta muy útil para comprender el modo de funcionamiento del resto de los autómatas secuenciales.

Se iniciará dicho estudio mediante el análisis de un sistema ya construido adecuadamente elegido, lo cual permitirá observar todos los fenómenos que pueden presentarse en un sistema de este tipo.

En la figura 6.7 se representa el esquema del circuito secuencial asíncrono cuyo análisis se va a realizar. Posee dos variables de entrada x_1 y x_2, dos variables de estado interno Y_1 e Y_2 y dos variables de salida z_1 y z_2 obtenidas mediante combinación de las variables de entrada y las variables de estado interno. Se trata, por tanto, de un autómata de Mealy.

Para simplificar la nomenclatura de las variables se designan por y las variables de estado interno en el instante $t(Y_t)$ y por Y las mismas variables en el instante $t + 1(Y_{t+1})$.

Del circuito de la figura 6.7 se deducen las expresiones algebraicas de Y_1, Y_2, z_1 y z_2 que son:

$$Y_1 = \overline{\overline{y_1\,\bar{y}_2} \quad \overline{\bar{x}_1\,x_2} \quad \overline{\bar{y}_1\,y_2} \quad \overline{x_1\,\bar{x}_2}}$$

$$Y_2 = \overline{\overline{\bar{x}_1\,x_2} \quad \overline{\bar{y}_1\,y_2} \quad \overline{x_1\,\bar{y}_1} \quad \overline{x_2\,\bar{y}_2}}$$

$$z_1 = \overline{\overline{\bar{y}_1\,x_2} \quad \overline{\bar{x}_1\,x_2}}$$

$$z_2 = \overline{y_1\,\bar{y}_2 + x_1}$$

Aplicando el teorema de De Morgan se obtienen finalmente las expresiones de las variables de estado interno y las variables de salida en forma de suma de productos lógicos:

$$Y_1 = y_1\,\bar{y}_2 + \bar{x}_1\,x_2 + \bar{y}_1\,y_2\,(\bar{x}_1 + x_2)$$
$$Y_2 = \bar{x}_1\,x_2 + \bar{y}_1\,y_2 + x_1\,\bar{y}_1 + x_2\,\bar{y}_2$$
$$z_1 = \bar{y}_1\,x_2 + \bar{x}_1\,x_2$$
$$z_2 = \bar{x}_1\,(\bar{y}_1 + y_2) = \bar{x}_1\,\bar{y}_1 + \bar{x}_1\,y_2$$

Partiendo de estas ecuaciones algebraicas se puede realizar una tabla de Karnaugh de cada una de las funciones Y_1, Y_2, z_1 y z_2 en función de las variables binarias de las cuales dependen, que son y_1, y_2, x_1 y x_2. Para ello solo hay que observar el valor lógico que toma cada función para todas las combinaciones de las variables.

A continuación se representan las variables de estado interno Y_1 e Y_2 en una sola tabla y las variables de salida z_1 y z_2 en otra.

La primera de estas tablas (figura 6.8) se llama tabla de excitación porque indica el nuevo estado interno del sistema a partir de un estado de entrada y el estado interno anterior.

La segunda tabla se representa en la figura 6.9 y recibe el nombre de matriz de salida porque indica el estado de salida correspondiente a cada combinación de estado interno y estado de entrada.

Observando la matriz de excitación de la figura 6.8 se comprueba la existencia de estados internos estables o inestables. Un estado interno es estable para un determinado estado de las variables de entrada, si el estado interno siguiente coincide con él. En caso contrario dicho estado interno es inestable. Por ejemplo, se puede observar que el estado interno $y_1y_2 = 00$ es estable cuando el estado de entrada es $x_1x_2 = 00$ porque el estado interno siguiente es 00. Por el contrario, el estado interno $y_1y_2 = 00$ no es estable cuando el estado de entrada es 01 porque el nuevo estado interno es 11, distinto de 00. Partiendo de la matriz de excitación del sistema se puede observar la evolución del mismo y representarla en un gráfico que se denomina de funcionamiento o evolución del sistema. Dicho gráfico, representado en la figura 6.10, se obtiene partiendo de la matriz de excitación e indicando cada estado interno estable con la letra E y cada estado interno inestable con la letra I, y observando la evolución entre estados internos para cada estado de entrada. Por ejemplo, en la figura 6.8 se observa que, cuando el estado de entrada es 00, del estado interno 01 se pasa al 11 y de éste al 00 que es estable; el paso de un estado a otro se indica mediante una flecha.

y_1y_2 \ x_1x_2	00	01	11	10
00	00	11	01	01
01	11	11	11	01
11	00	11	00	00
10	10	11	11	10

Y_1, Y_2

FIGURA 6.8.—**Matriz de excitación.**

y_1y_2 \ x_1x_2	00	01	11	10
00	01	11	10	00
01	01	11	10	00
11	01	11	00	00
10	00	10	00	00

z_1, z_2

FIGURA 6.9.—**Matriz de salida.**

En el gráfico de funcionamiento de la figura 6.10 se pueden estudiar las características y fenómenos más importantes de los sistemas secuenciales asíncronos. Estos se caracterizan, en general, por encontrarse en un estado interno estable para un cierto estado de entrada y, al producirse un cambio de éste, el sistema evoluciona hacia un nuevo estado estable.

Puede suceder que para un determinado estado de entrada no exista ningún estado interno estable. Si esto sucede, el sistema evolucionará continuamente entre estados inestables y, dado que el número de éstos es finito, acabará finalmente por recorrer un ciclo. En la tabla de la figura 6.8 se observa que, si a partir del estado interno 11, estable para el estado de entrada 01, éste pasa a 11, el sistema comienza a realizar un ciclo entre los estados 00, 01 y 11. Si, cuando el sistema está evolucionando, en un ciclo se produce un cambio del estado de entrada, es imposible, en general, predecir el estado estable final, el cual depende del estado interno en que se encuentre el sistema en el momento de producirse la variación. Por ejemplo, si

el sistema está en el interior del ciclo correspondiente al estado de entrada 11 y éste pasa a ser 01, el sistema pasa al estado interno 01 si se encuentra en el 01 en el ciclo y, por el contrario, pasa al 11 si se encuentra en este estado.

De todo lo dicho se deduce que se han de evitar los ciclos en los sistemas secuenciales asíncronos.

Otro fenómeno a estudiar es el de las transiciones entre estados no adyacentes (que se denominan simplemente transiciones no adyacentes). Estas transiciones pueden presentarse tanto entre estados de entrada como entre estados internos. Recuérdese que dos combinaciones binarias no son adyacentes cuando de una a otra cambia de estado más de una variable binaria.

En un sistema físico es imposible que más de una variable binaria cambie simultáneamente y, por tanto, el cambio entre dos estados no adyacentes se realiza siempre pasando por un estado intermedio. Esto hace que la transición entre estados no adyacentes pueda ser crítica o no crítica. Una transición entre estados no adyacentes no es crítica cuando el estado final alcanzado después de la transición es el mismo, independientemente de los estados intermedios a través de los cuales ésta se produzca, e igual al estado que el sistema alcanzaría en caso de que todas las variables cambiasen simultáneamente.

Por ejemplo, la transición entre los estados internos 00 y 11 correspondientes al estado de entrada 01 (columna 2 de la tabla de funcionamiento de la figura 6.10) no es crítica porque, tanto si el cambio se produce pasando por el estado 10 como por el 01, el estado estable final resultante es el mismo (11).

La transición entre estados no adyacentes es crítica cuando el estado final alcanzado al producirse la misma a través de un estado intermedio no es el mismo que si todas las variables cambiasen simultáneamente. Por ejemplo, la transición entre los estados internos 00 y 11 correspondientes al estado de entrada 00 (columna 1 de la tabla de funcionamiento de la figura 6.10) es crítica porque si del estado 11 se pasa primero al 10, se quedará en dicho estado, que es estable, y no alcanzará nunca el estado 00.

Se deduce, pues, que en el diseño de todo sistema secuencial asíncrono se han de evitar las transiciones no adyacentes críticas. Debido a que no es posible *a priori* saber si una transición no adyacente será o no crítica, se ha de evitar que al producirse un cambio del estado de la entrada o del estado interno, lo haga más de un bit simultáneamente.

A partir de las tablas de excitación y de salida en las que a cada línea le corresponde un estado interno codificado mediante una combinación binaria, se puede continuar el análisis hasta llegar al planteamiento inicial del problema. Para ello, a cada estado interno se le asigna un número decimal y se representa en una misma tabla el estado interno siguiente y el estado de salida que le corresponde. Se obtiene de esta forma la tabla de flujo, denominada así porque representa la evolución global del sistema. Dicha tabla se indica en la figura 6.11.

El paso inverso, es decir, de la tabla de flujo a las matrices de excitación y de salida, consiste en la codificación de los estados internos, que es uno de los problemas más importantes de la síntesis de los sistemas secuenciales asíncronos de realimentación directa porque de él depende la complejidad del sistema y además se ha de realizar evitando las transiciones no adyacentes.

y_1y_2 \ x_1x_2	00	01	11	10
00	E	I	I	I
01	I	I	I	E
11	I	E	I	I
10	E	I	I	E

FIGURA 6.10.—**Tabla de funcionamiento.**

	x_1x_2 00	01	11	10
1	1/01	3/11	2/10	2/00
2	3/01	3/11	3/10	2/00
3	1/01	3/11	1/00	1/00
4	4/00	3/10	3/00	4/00

FIGURA 6.11.—**Tabla de flujo.**

La tabla de flujo representa, no solamente los estados inicial y final del sistema en una evolución, sino las transiciones o pasos intermedios. Por ejemplo, si el sistema se encuentra en el estado interno estable 2 correspondiente al estado de entrada 10 (columna 4 de la tabla de flujo de la figura 6.11) y éste conmuta al 00, el sistema pasa del estado 2 al 3 y de éste al 1 en el cual permanece por ser un estado estable.

La información contenida en la tabla de flujo se puede representar en un diagrama de flujo como el indicado en la figura 6.12 en el cual se asigna un punto del plano a cada estado interno y se indican mediante flechas las transiciones entre

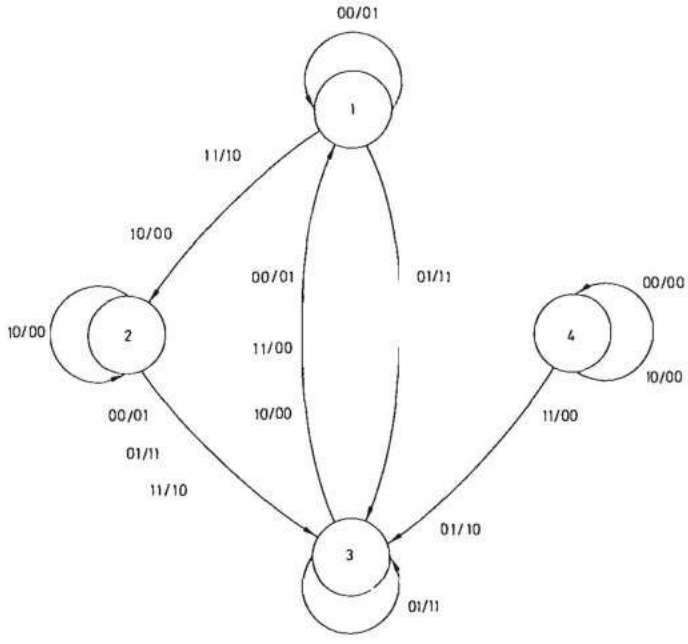


FIGURA 6.12.—**Diagrama de flujo.**

estados, asignándoles los estados de entrada y salida correspondientes a cada transición.

En un sistema asíncrono lo que resulta realmente importante son los estados finales estables y la transición entre ellos, independientemente de los estados intermedios que se recorran. Por eso partiendo de la tabla de flujo se puede llegar a otra en la que simplemente se indiquen los estados estables, asignándoles un número decimal encerrado en un círculo y las transiciones entre ellos se indican mediante el mismo número del estado estable final al cual ha de pasar el sistema. Si en una columna no existe ningún estado estable, se indican mediante un guión todos los estados de la misma.

En la tabla de la figura 6.11 se observa que existen cinco estados estables que se enumeran del 1 al 5. En la primera, segunda y tercera filas existe un estado estable en cada una que se denominan respectivamente 1, 2 y 3. En la cuarta fila existen dos estados estables que se denominan 4 y 5. De esta forma se obtiene la tabla de la figura 6.13 que se denomina tabla de fases y que es la primera que se realiza en la síntesis de un sistema secuencial asíncrono. En ella los estados estables se indican encerrándolos en un círculo y a los estados inestables se les asigna el número del estado estable al cual se va.

X_1X_2	00	01	11	10
	(1)	3	–	2
	1	3	–	(2)
	1	(3)	–	2
	(5)	3	–	(4)

FIGURA 6.13.—**Tabla de fases.**

Las especificaciones de un sistema asíncrono indican los estados de salida para todas las secuencias admisibles de los estados de entrada. Estas especificaciones se pueden representar gráficamente en un diagrama de secuencia en el cual se indican las transiciones entre los diferentes estados de entrada y el estado de salida correspondiente. Dichas transiciones se representan en una tabla de fases en la que se indica además el estado de salida correspondiente a cada estado interno. En esta tabla de fases inicial se indica un solo estado estable por línea. Una vez realizada esta tabla de fases es necesario estudiar su posible reducción a fin de lograr la realización del sistema propuesto con un número mínimo de estados internos.

La reducción de una tabla de fases se realiza mediante la fusión de dos o más líneas de la misma en una sola línea.

La fusión de una tabla de fases es un paso importante porque el número de estados internos, será igual al de líneas y el sistema será tanto más sencillo cuanto menor sea el número de éstas. Para estudiar la fusión de líneas de una tabla de fases hay que tener en cuenta el concepto de estados equivalentes y estados pseudoequivalentes. El desarrollo de estos conceptos presenta un interés didáctico pero sus dificultades de aplicación le quitan utilidad práctica.

En efecto la aplicación práctica de la tabla de fases al diseño de sistemas secuenciales asíncronos resulta difícil, por no decir imposible, en cuanto el número de variables de entrada es superior a dos y las secuencias a memorizar mayor también de dos.

Por ello carece de interés práctico estudiar los métodos de fusión de las tablas de fases. Al lector interesado se le remite a la bibliografía [ALDA 80] [HOLD 84] [MAND 87].

No obstante, es útil el análisis de un ejemplo sencillo que da como resultado uno de los sistemas secuenciales asíncronos de uso más generalizado.

Ejemplo 6.1: Diseñar un sistema secuencial asíncrono con dos variables de entrada x_1 y x_2 y una variable de salida Z que ha de actuar de acuerdo con las especificaciones siguientes:

a) Si la variable x_1 toma el valor lógico uno, la salida Z del sistema pasa al estado lógico uno independientemente del estado en que se encuentre y continúa en él aunque la variable x_1 vuelva al estado lógico cero. Sucesivas conmutaciones de la variable x_1 no cambian el estado de la salida.

b) Si la variable x_2 toma el valor lógico uno, la salida Z del sistema pasa al estado lógico cero independientemente del estado en que se encuentre anteriormente y continúa en él aunque la variable x_2 vuelva al estado lógico cero. Sucesivos cambios de estado de x_2 no modifican el estado de la salida.

c) Las variables x_1 y x_2 no pueden encontrarse en estado uno simultáneamente ni cambiar de estado en el mismo instante.

Estas especificaciones se pueden representar en un diagrama de secuencia como el de la figura 6.14 en el cual se observa que el sistema actúa de acuerdo con aquéllas. A cada combinación de estados de entrada y salida diferentes se le asigna un número decimal que corresponde a un estado total del sistema determinado.

Observando las transiciones entre estados en el diagrama de secuencia, resulta sencillo obtener la tabla de fases representada en la figura 6.15. En esta tabla se indican mediante un guión aquellos estados a los cuales no llega nunca el sistema de acuerdo con las especificaciones expuestas. Por ejemplo, si el sistema se encuentra en el estado (3) estable con la entrada en estado 10, para que ésta pase al estado 01 han de cambiar simultáneamente las dos entradas, lo cual se ha especificado que no puede suceder. Por ello se ha colocado un guión en la casilla correspondiente.

A cada estado estable se le asigna una línea de la tabla y se indican las transiciones a partir de él. Por ejemplo, si se llama (1) al estado del sistema en que la entrada es 00 y la salida 0, mediante cambio del estado de entrada a 01 o 10 se puede poner el sistema en dos nuevos estados estables (2) y (3), el primero

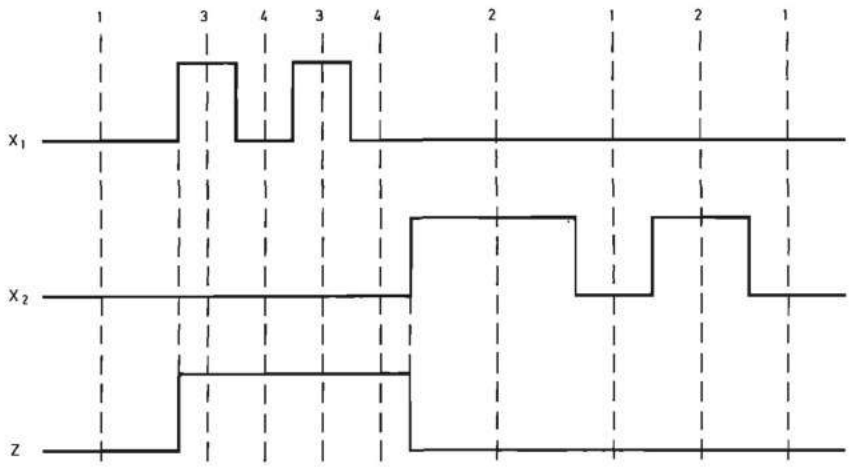


FIGURA 6.14.—**Diagrama de secuencia temporal de las relaciones entre las variables de entrada y las variables de salida del sistema secuencial asíncrono del ejemplo 6.1.**

$X_1 X_2$	00	01	11	10	Z
A	①	2	–	3	0
B	1	②	–	–	0
C	4	–	–	③	1
D	④	2	–	3	1

FIGURA 6.15.—**Tabla de fases del sistema secuencial asíncrono del ejemplo 6.1.**

con salida 0 y el segundo con salida 1, a los que se les asigna una nueva línea. Las transiciones de ① a ② y ③ se indican mediante los estados inestables 2 y 3 respectivamente.

Pero en la tabla de la figura 6.15, se pueden fusionar las líneas *A* y *B* en una sola representada en la figura 6.16. De igual forma las líneas *C* y *D* se pueden fusionar en otra. El lector puede comprobar que la tabla de la figura 6.16 contiene la misma información que la 6.15. En efecto en la primera línea se indica que el sistema pasa del estado ① al ② si las entradas x_1, x_2 cambian de 00 a 01 y viceversa si pasan de 01 a 00. De igual forma se indica que el sistema pasa del estado ① al ③ a través del estado inestable 3 si x_1, x_2 pasan de 00 a 10.

En la tabla de fases de la figura 6.16 hay sólo dos líneas, a cada una de las cuales le corresponde un estado de salida. El sistema obtenido es, por lo tanto, un

X_1X_2	00	01	11	10	Z
	①	②	–	3	0
	④	2	–	③	1

FIGURA 6.16.—**Tabla de fases reducida obtenida a partir de la tabla de la figura 6.15.**

X_1X_2	00	01	11	10
1	1/0	1/0	–	2/0
2	2/1	1/1	–	2/1

FIGURA 6.17.—**Tabla de flujo del sistema secuencial asíncrono del ejemplo 6.1.**

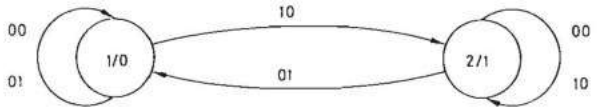


FIGURA 6.18.—**Diagrama de flujo del sistema secuencial asíncrono del ejemplo 6.1.**

autómata de Moore, del que se puede realizar la tabla de flujo y el diagrama de flujo que se representan respectivamente en las figuras 6.17 y 6.18.

La codificación de una tabla de fases de dos líneas no presenta ningún problema porque para realizarla solamente se necesita una variable de estado interno que proporciona los dos estados necesarios y es imposible la existencia de transiciones críticas entre estados internos no adyacentes.

Se asigna el estado interno 0 a la primera línea y el 1 a la segunda línea. La matriz de excitación se realiza asignando a cada estado estable el estado interno de su línea y, a cada estado inestable, el estado interno del estado estable correspondiente. En la figura 6.19 se representa la matriz de excitación obtenida y en la figura 6.20 la matriz de salida. De esta última se deduce:

$$Z = y$$

X_1X_2 / y	00	01	11	10
0	0	0	X	1
1	1	0	X	1

Y

FIGURA 6.19.—**Matriz de excitación del sistema secuencial asíncrono del ejemplo 6.1.**

X_1X_2 / y	00	01	11	10
0	0	0	X	0
1	1	1	X	1

Z

FIGURA 6.20.—**Matriz de salida del sistema secuencial asíncrono del ejemplo 6.1.**

La expresión algebraica de Y se obtiene a partir de la tabla de Karnaugh que coincide con la matriz de excitación (fig. 6.19), porque tiene una sola variable de estado interno. Según que a los dos estados indiferentes se les asigne el valor cero o el uno, se obtienen dos expresiones algebraicas diferentes. Asignándoles el valor uno la expresión resultante es:

$$Y = x_1 + \bar{x}_2\, y$$

que realizada con puertas NO-Y (NAND) resulta:

$$Y = \overline{\overline{x_1 + \bar{x}_2 y}} = \overline{\bar{x}_1 \bar{x}_2 y}$$

En la figura 6.21 se representa el esquema correspondiente.
Asignando el valor cero a los términos indiferentes, resulta:

$$Y = x_1 \bar{x}_2 + \bar{x}_2 y = \bar{x}_2 (x_1 + y)$$

que realizada con puertas NO-O (NOR) resulta:

$$Y = \overline{\overline{\bar{x}_2 (x_1 + y)}} = \overline{x_2 + \overline{x_1 + y}}$$

En la figura 6.22 se representa el esquema del sistema resultante.

La única diferencia entre ambas realizaciones es su comportamiento cuando las entradas x_1 y x_2 toman simultáneamente el valor uno (lo cual se especificó como estado de entrada imposible). El sistema de la figura 6.21 es de grabación prioritaria porque si x_1 y x_2 toman el valor uno al mismo tiempo, la salida toma el valor uno y el de la figura 6.22 es de borrado prioritario porque en dicha circunstancia la salida toma el valor cero.

Los sistemas secuenciales que poseen una sola variable de estado interno se denominan biestables porque poseen dos estados estables. El que ahora se acaba de diseñar recibe el nombre de biestable *R-S* (Reset-Set). A las entradas x_1 y x_2 se les asignan, respectivamente, las letras *S* y *R*.

Los biestables *R-S* constituyen células básicas que por su sencillez pueden ser utilizadas como bloque funcional para formar cualquier sistema secuencial asíncrono tal como se estudia más adelante.

De ahí el interés de que se crease un símbolo normalizado para representarlos.

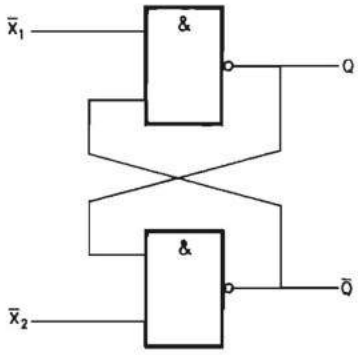


FIGURA 6.21.—**Biestable *R-S* de grabado prioritario.**

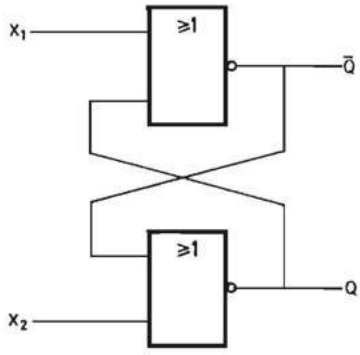


FIGURA 6.22.—**Biestable *R-S* de borrado prioritario.**

FIGURA 6.23.—**Símbolo lógico normalizado del biestable *R-S*: *a*) Correspondiente al circuito de la figura 6.21; *b*) Correspondiente al circuito de la figura 6.22.**

En la figura 6.23*a* se representa el símbolo correspondiente al biestable de la figura 6.21 y en la 6.23*b* el correspondiente al biestable de la figura 6.22. En los símbolos de la figura 6.23 no se indica cuál es la respuesta del biestable cuando ambas entradas *S* y *R* se encuentran en estado uno. Es posible hacerlo añadiéndoles un símbolo numérico a las letras *S* y *R* y colocando el mismo dígito en la salida adecuada. Por ejemplo, si a *S* se le añade el número 1 (*S*1), la salida indicada con 1 se pone a uno cuando *S* se coloca en dicho nivel. De igual forma si a *R* se le añade el número 2 (*R*2), la salida indicada con 2 se pone a cero cuando *R* se coloca en uno.

De acuerdo con el convenio descrito, en la figura 6.24*a* se representa el biestable *R-S* de grabado prioritario y en la 6.24*b* el de borrado prioritario.

Como ya se indicó anteriormente, resulta difícil en la práctica la realización de la tabla de fases a partir de las especificaciones de funcionamiento de un autómata asíncrono. Por otra parte, los métodos de codificación de los estados internos son de difícil aplicación para evitar la aparición de transiciones críticas.

Por todo ello no se insiste más en el estudio de los sistemas secuenciales asíncronos caracterizados por niveles con reacción directa. El lector interesado en los mismos puede dirigirse a la bibliografía [ALDA 80] [HOLD 84] [MCCL 86] [MAND 87] [UNGE 69] [UNGE 71].

No obstante, es interesante analizar los sistemas secuenciales asíncronos elementales activados por niveles que reciben el nombre de biestables y de los que el *R-S* es el ejemplo más característico. Por ello a los biestables asíncronos activados por niveles se dedica el apartado siguiente.

FIGURA 6.24.—**Símbolo lógico normalizado del biestable *R-S*: *a*) Con grabado prioritario; *b*) Con borrado prioritario.**

6.2.2.2 Biestables asíncronos activados mediante niveles. Los biestables asíncronos activados por niveles son sistemas secuenciales asíncronos que poseen una sola variable de salida y cuyas entradas controlan directamente su estado cuando alcanzan determinados niveles lógicos activos.

S	R	y_t	$y_{t+1} = Y$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

TABLA 6.3.—**Tabla de excitación del biestable *R-S*.**

S	R	y_{t+1}
0	0	y_t
0	1	0
1	0	1
1	1	X

TABLA 6.4.—**Tabla de excitación del biestable *R-S*.**

Las características de cada uno de estos biestables se describen mediante una tabla de excitación, que es una tabla de verdad en la que se indica el valor del estado interno $Y = y_{t+1}$ en el instante $t + 1$, en función de los estados de las variables de entrada y del estado interno en el instante t.

El más elemental de todos estos sistemas es el biestable *R-S* cuyo diseño ha sido realizado en el apartado 6.2.2.1. Sus tablas de excitación se representan en las tablas 6.3 y 6.4. Esta última es una forma reducida de representar la forma de actuación del sistema. En este tipo de biestable se limita la actuación de las entradas especificando que nunca pueden ser ambas activas simultáneamente. Por tanto, en esa situación el estado interno y_{t+1} no está definido. En las tablas 6.3 y 6.4 se supone que el nivel activo es el uno lógico.

Este biestable se representa en las figuras 6.21 y 6.22 y es el más sencillo de todos.

Otros biestables que actúan mediante niveles son:

a) Biestable *J-K:* Su diferencia con respecto al *R-S* es que admite que ambas entradas estén en estado uno simultáneamente y, en esta situación de la entrada,

J	K	y_t	y_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

TABLA 6.5.—**Tabla de excitación del biestable *J-K*.**

J	K	y_{t+1}
0	0	y_t
0	1	0
1	0	1
1	1	$\bar{y}_t$

TABLA 6.6.—**Tabla de excitación del biestable *J-K*.**

P	Q	y_t	y_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

TABLA 6.7.—**Tabla de excitación del biestable *P-Q*.**

P	Q	y_t
0	0	y_t
0	1	0
1	0	1
1	1	y_t

TABLA 6.8.—**Tabla de excitación del biestable *P-Q*.**

el estado interno del biestable debe cambiar. Este biestable no se utiliza mucho en la práctica porque, cuando ambas entradas se encuentran en estado uno simultáneamente, se convierte en un oscilador.

Por tanto, la tabla de excitación del biestable *J-K* es la indicada en las tablas 6.5 y 6.6.

b) Biestable *P-Q:* Su diferencia con respecto al *R-S* es que se admite que las dos entradas estén en estado lógico uno simultáneamente y en dicha situación el estado interno permanece invariable. La tabla de excitación se representa en las tablas 6.7 y 6.8.

c) Biestable *T:* Este biestable posee una sola entrada *T* que, cada vez que adopta el nivel lógico uno, hace que cambie de estado la salida y para dicho valor de la entrada *T* también se convierte en un oscilador.

T	y_t	y_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

TABLA 6.9.—**Tabla de excitación del biestable *T*.**

T	y_{t+1}
0	y_t
1	$\bar{y}_t$

TABLA 6.10.—**Tabla de excitación del biestable *T*.**

En las tablas 6.9 y 6.10 se representa su tabla de excitación.

6.2.2.3 Síntesis de los sistemas secuenciales asíncronos con realimentación mediante biestables. La importancia práctica de los biestables asíncronos activados por niveles es debida a que pueden ser utilizados como bloques básicos para la construcción de sistemas secuenciales asíncronos más complejos.

De todos los estudiados en el apartado anterior, el biestable más utilizado es el *R-S* en cualquiera de sus dos versiones, con grabación prioritaria o con borrado prioritario.

El diagrama de bloques de un sistema secuencial asíncrono realizado con biestables *R-S* es el de la figura 6.25, que constituye un caso particular de la figura 6.5. Este sistema evoluciona de una forma autónoma. Dado un estado de entrada X_h (un determinado estado de todas las variables $x_1, x_2, \ldots, x_n$) y un estado interno Q_j (un determinado estado de todas las salidas de los biestables $Q_1, Q_2, \ldots, Q_m$) estable para dicho estado de entrada, los valores de las variables S_i y R_i (para i de 1 a n) toman un valor tal que los biestables *R-S* permanecen en un estado fijo.

Al cambiar una variable de entrada, el sistema ha de ir a un nuevo estado interno que puede ser distinto del actual. En este caso el sistema combinacional hace que para el nuevo estado de entrada y el estado interno actual cambien las S_i y R_i adecuadas para que se produzca el nuevo estado interno. Si este nuevo estado interno es estable, las entradas S_i y R_i permanecen invariables o se mantienen en nivel cero (de manera que no cambie el estado) hasta que se produzca un nuevo cambio en el vector de entrada. Si el nuevo estado interno no es estable, se produce un nuevo cambio en las entradas S_i y R_i y así sucesivamente hasta alcanzar un estado interno estable.

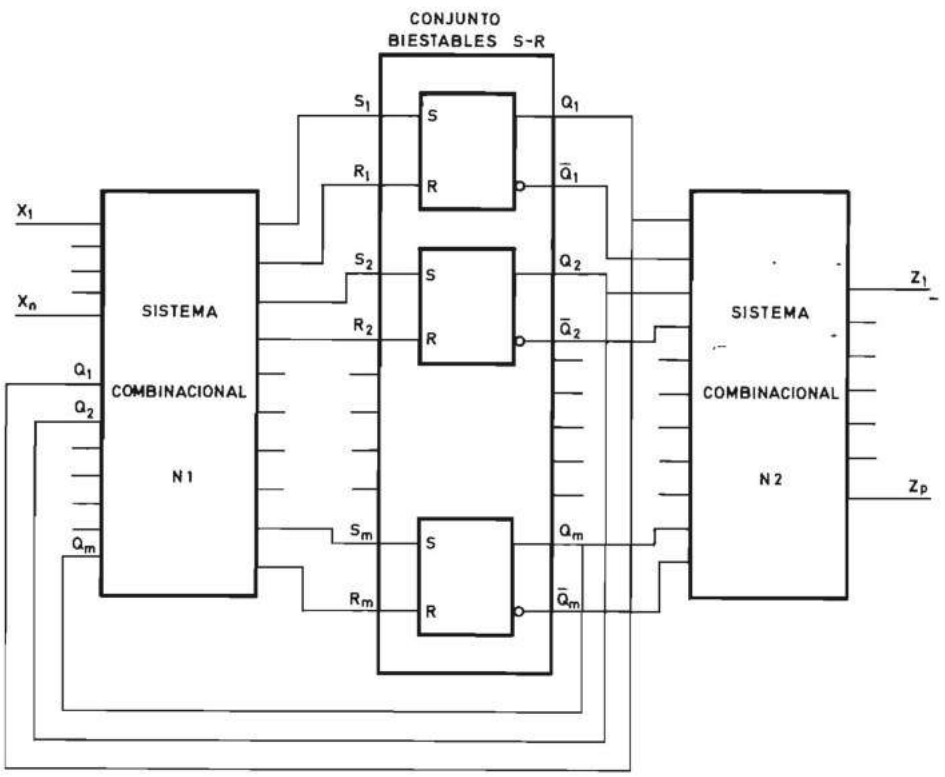


FIGURA 6.25.—**Diagrama de bloques de un sistema secuencial asíncrono realizado con biestables *R-S*.**

El sistema asíncrono evoluciona, por lo tanto, entre estados internos de una forma autónoma cuando se produce un cambio de las entradas. Su forma de operación se puede representar mediante un grafo denominado diagrama de flujo como el indicado en la figura 6.26, que representa el diagrama de flujo de un autómata asíncrono con cuatro estados estables E_1 a E_4. Las evoluciones entre estados se realizan al aparecer determinadas combinaciones de las variables de entrada o vectores de entrada. Por ejemplo, si el autómata se encuentra en el estado E_2 y aparece la combinación de entrada X_j, se produce el paso al estado E_3. Un ejemplo práctico del diagrama de flujo de un autómata asíncrono es el representado en la figura 6.12.

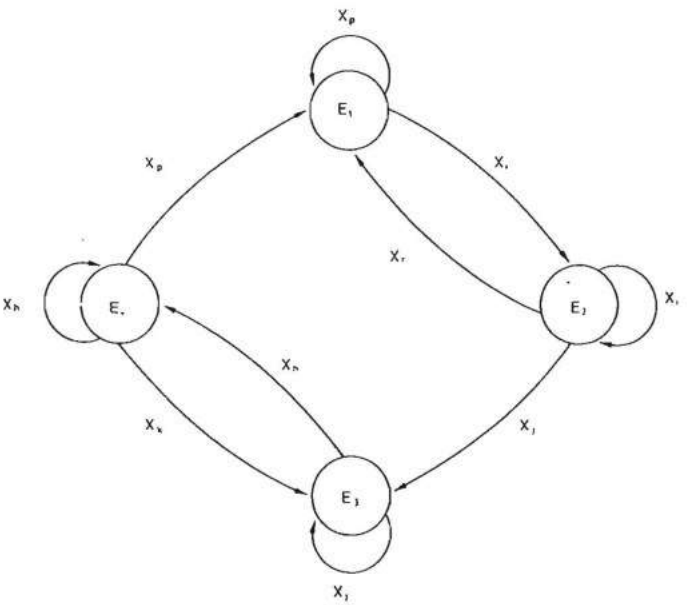


FIGURA 6.26.—**Diagrama de flujo de un autómata asíncrono.**

Es necesario recordar que cada una de las variables de estado interno Y_i de un sistema secuencial depende de las variables de entrada $X(x_1, x_2, \ldots, x_n)$ y las variables de estado interno $Y(y_1, y_2, \ldots, y_m)$, es decir:

$$Y_i = f(x_1, \ldots, x_n, y_1, \ldots, y_i, \ldots, y_m)$$

Los productos canónicos que forman parte de Y_i se pueden dividir en dos grupos según la variable y_i esté en forma directa o inversa. Sacando factor común y_i e $\bar{y}_i$ en cada uno de dichos grupos y denominando a las expresiones resultantes M_1 y P_1 se obtiene:

$$Y_i = M_1 y_i + P_1 \bar{y}_i$$

De igual forma, realizando lo mismo para la expresión canónica de sumas de productos de $\bar{Y}_i$ resulta:

$$\bar{Y}_i = P_0 y_i + M_0 \bar{y}_i$$

En la ecuación de Y_i los términos que forman parte de M_1 (mantener estado uno) son los correspondientes al estado de todas las variables de entrada y de estado interno menos y_i, para los cuales, siendo el estado interno i un uno lógico, debe de continuar en el mismo. Los términos que forman parte de P_1 (poner en estado uno) son aquellos para los que el estado interno i debe pasar de cero a uno.

De igual forma, en la ecuación de $\bar{Y}_i$ los términos que forman parte de P_0 (poner en cero) son los correspondientes a aquellos estados en los que $\bar{Y}_i$ debe pasar de cero a uno, es decir, Y_i de uno a cero y los términos de M_0 corresponden a los estados en que $\bar{Y}_i$ debe mantenerse en estado uno, es decir, Y_i en estado cero.

En la tabla de excitación de todo sistema secuencial se pueden sustituir los valores de Y_i por P_1, M_1, P_0 o M_0 según Y_i pase de 0 a 1, se mantenga en 1, pase de 1 a 0, o se mantenga en 0, respectivamente. De esta forma, se define el comportamiento del sistema por medio de sus variaciones y la tabla obtenida se denomina tabla de transiciones.

La aplicación práctica de este método presenta la dificultad de obtención de la tabla de excitación a partir de la tabla de fases ya expuesta anteriormente.

Por ello se va a utilizar solamente en un caso sencillo en el que la codificación de la tabla de fases no presenta ningún problema.

Se trata de un sistema secuencial asíncrono con una única variable de entrada denominada t y una variable de salida z. Su diagrama de evolución en el tiempo se representa en la figura 6.27. Cada vez que t cambia de uno a cero la salida z cambia de estado. Se define como un biestable porque posee una sola variable de salida, pero cuyos cambios de estado son debidos a los cambios de nivel o flancos de la variable de entrada en lugar de a los niveles. Como se verá más adelante los biestables asíncronos activados por flancos constituyen bloques funcionales aplicables a la síntesis de cualquier sistema secuencial asíncrono.

En el diagrama de evolución en el tiempo del sistema de la figura 6.27 se pueden observar los diferentes estados internos. Al estado en que se encuentra el sistema cuando su salida z está en estado cero al igual que la entrada t, se le denomina estado 1. Cuando t pasa al estado uno, la salida se mantiene en estado cero y el

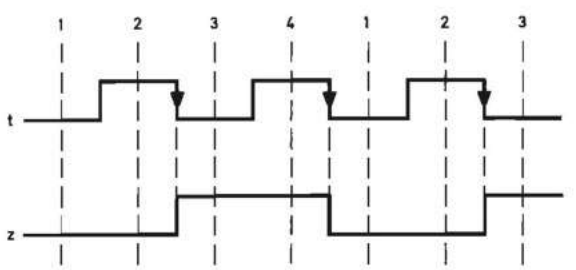


FIGURA 6.27.—**Diagrama de secuencia temporal que relaciona la variable de entrada (t) y la de salida (z) de un biestable t activado por flancos de bajada.**

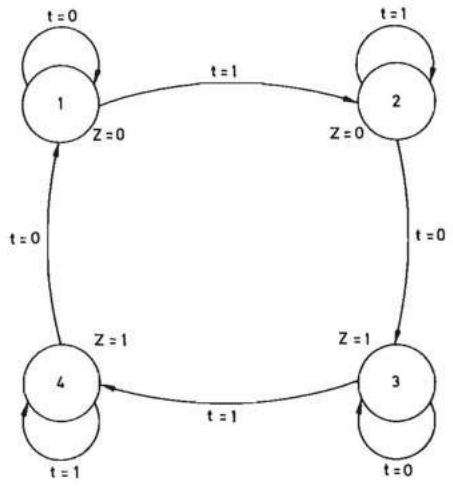


FIGURA 6.28.—**Diagrama de flujo del biestable *t* activado por flancos.**

t	0	1	Z
	(1)	2	0
	3	(2)	0
	(3)	4	1
	1	(4)	1

FIGURA 6.29.—**Tabla de fases del biestable *t* activado por flancos.**

sistema pasa al estado 2. Cuando la entrada t vuelve al estado cero, la salida pasa de cero a uno y el sistema alcanza el estado 3. Una nueva subida de t de cero a uno hace que la salida z continúe en uno y el sistema conmuta del estado 3 al 4. Finalmente, un nuevo cambio de t de uno a cero produce el comienzo de un nuevo ciclo. Lo expresado se puede representar gráficamente mediante el diagrama de flujo de la figura 6.28. De este diagrama se obtiene fácilmente la tabla de fases de la figura 6.29.

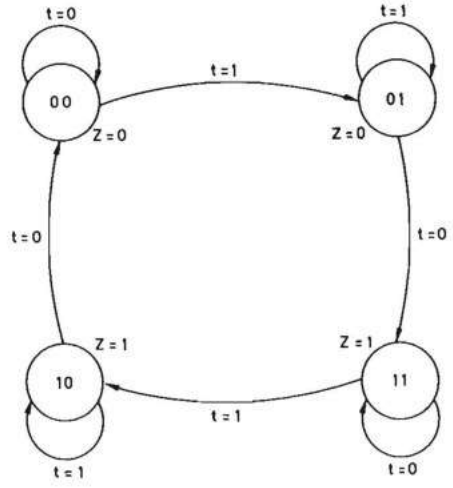


FIGURA 6.30.—**Diagrama de flujo codificado del biestable *t* activado por flancos.**

$y_1\ y_2$ \ t	0	1	Z
00	00	01	0
01	11	01	0
11	11	10	1
10	00	10	1

FIGURA 6.31.—**Tabla de excitación del biestable *t* activado por flancos.**

Observando el diagrama de estado o la tabla de fases, se llega a la conclusión de que no pueden aparecer transiciones críticas si se asigna a cada estado un código binario adyacente al de los estados contiguos. De acuerdo con ello, si a los estados 1, 2, 3 y 4 se les asignan respectivamente los códigos 00, 01, 11 y 10 se obtiene el diagrama de flujo de la figura 6.30 y la tabla de excitación de la figura 6.31.

A partir de la tabla de excitación se obtiene la tabla de transiciones de la figura 6.32 aplicando la teoría desarrollada para los términos P_1, M_1, P_0 y M_0.

y_1y_2 \ t	0	1
00	$M_0\,M_0$	$M_0\,P_1$
01	$P_1\,M_1$	$M_0\,M_1$
11	$M_1\,M_1$	$M_1\,P_0$
10	$P_0\,M_0$	$M_1\,M_0$

$Y_1.Y_2$

FIGURA 6.32.—**Tabla de transiciones del biestable *t* activado por flancos.**

Por ejemplo, para $t = y_1 = y_2 = 0$, el nuevo estado interno Y_2 es 0, es decir, el término correspondiente pertenece a M_0 (mantener en estado cero). De igual forma se obtiene el resto de la tabla.

Para realizar un circuito secuencial con cualquier biestable de los estudiados, es necesario saber qué valores lógicos han de ser aplicados a sus entradas para cada una de las transiciones, lo cual se deduce a partir de su tabla de excitación.

En la tabla 6.11 se repite la tabla de excitación del biestable *R-S* y se indica la transición correspondiente a cada combinación de variables de la tabla.

S	R	y_t	y_{t+1}	*Transición*
0	0	0	0	M_0
0	0	1	1	M_1
0	1	0	0	M_0
0	1	1	0	P_0
1	0	0	1	P_1
1	0	1	1	M_1
1	1	0	X	
1	1	1	X	

TABLA 6.11.—**Tabla de excitación del biestable *R-S*.**

Y	P_1	M_1	P_0	M_0
S	1	X	0	0
R	0	0	1	X

TABLA 6.12

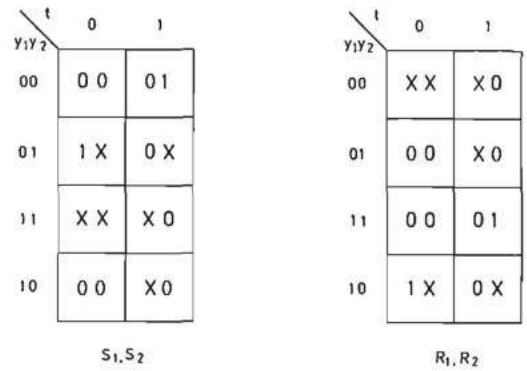


FIGURA 6.33.—**Tablas de verdad de las variables S_1, S_2, R_1 y R_2 del biestable *t* activado por flancos.**

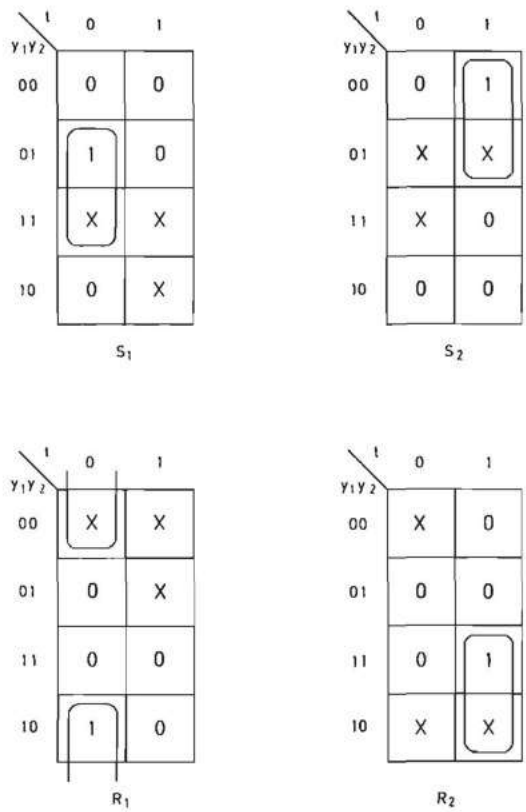


FIGURA 6.34.—**Tablas de Karnaugh de las variables S_1, S_2, R_1 y R_2 del biestable *t* activado por flancos.**

Partiendo de dicha tabla se deducen los valores que deben tomar las entradas S y R del biestable para cada una de las transiciones, lo cual se indica en la tabla 6.12. Por ejemplo, la transición P_1 (puesta a uno) se produce para $S = 1$ y $R = 0$ y el mantenimiento del estado uno M_1 para $S = 0$ o 1 y $R = 0$. De igual forma se deducen P_0 y M_0.

Para realizar la memoria t con biestables R-S se necesitan dos biestables para ambas variables de estado interno y es necesario deducir las ecuaciones de las entradas S y R de ambos.

Para lograrlo, se sustituyen las transiciones de la tabla de transición del sistema que se va a realizar por los valores lógicos correspondientes para S y R indicados en la tabla 6.12. De esta forma se obtienen las tablas de la figura 6.33. De ellas se deducen las tablas de Karnaugh de la figura 6.34 y se obtienen las expresiones:

$$S_1 = \bar{t}\, y_2; \qquad S_2 = t\, \bar{y}_1$$
$$R_1 = \bar{t}\, \bar{y}_2; \qquad R_2 = t\, y_1$$

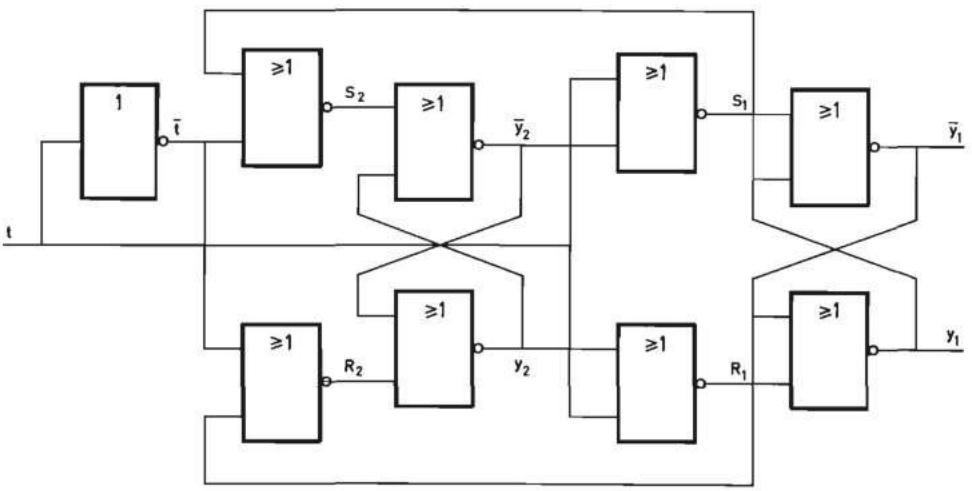


FIGURA 6.35.—**Esquema del biestable t activado por flancos.**

que, realizadas con puertas NO-O, resultan:

$$S_1 = \overline{t + \bar{y}_2}; \qquad S_2 = \overline{\bar{t} + y_1}$$
$$R_1 = \overline{\bar{t} + y_2}; \qquad R_2 = \overline{\bar{t} + \bar{y}_1}$$

Dado que $Z = y_1$, lo cual se deduce de la tabla de la figura 6.31, se obtiene el esquema del biestable t representado en la figura 6.35.

6.2.3 Sistemas secuenciales asíncronos caracterizados mediante cambios de nivel

En los apartados anteriores se han estudiado los sistemas secuenciales asíncronos caracterizados por medio de niveles. El método descrito adolece de los siguientes defectos que hacen muy difícil su aplicación práctica:

a) Dificultad de realización de la tabla de fases cuando el número de variables de entrada es superior a dos.

b) Facilidad de equivocación cuando el número de secuencias de las variables de entrada que ha de memorizar el sistema es elevado.

c) Posible aparición de transiciones críticas en la codificación.

d) Posibilidad de existencia de fenómenos aleatorios en la realización física.

A continuación se desarrolla un método que caracteriza a los sistemas secuenciales asíncronos por medio de los cambios de nivel de las variables de entrada y elimina prácticamente todos los inconvenientes descritos.

6.2.3.1 Especificaciones de transición de los sistemas secuenciales asíncronos caracterizados mediante cambios de nivel. En el estudio de los sistemas secuenciales asíncronos mediante niveles se crean estados internos estables para una determinada combinación de todas las variables de entrada y se supone que las transiciones entre estados internos se producen al aparecer una combinación diferente de todas las variables de entrada. Si se designan las variables de entrada por la letra x minúscula seguida de un subíndice, se pueden definir los vectores de entrada como un cierto estado de todas las variables de entrada:

$$X_a \equiv \{x_1^*, x_2^*, \ldots, x_i^*, \ldots, x_n^*\}$$

en el que x_i^* representa a la variable x_i en forma directa (x_i) o inversa ($\overline{x_i}$).

Por ejemplo, si las variables de entrada son x_1, x_2 y x_3, un vector de entrada X_a es:

$$X_a = \bar{x}_1 x_2 \bar{x}_3$$

El número total de vectores de entrada es 2^n, siendo n el número de variables de entrada.

En la figura 6.36*a* se representa gráficamente la especificación de cambio de estado interno en el método de niveles. Se supone al sistema secuencial asíncrono en el estado E_j estable y que permanece en él mientras existe el vector de entrada X_β. Si el vector de entrada X_β pasa a ser X_α, el autómata cambia del estado interno E_j al E_i y en él permanece mientras el vector de entrada es X_α. Se supone que X_α y X_β se diferencian en el estado de una sola variable, lo cual es equivalente a decir que solamente puede cambiar una variable de entrada simultáneamente.

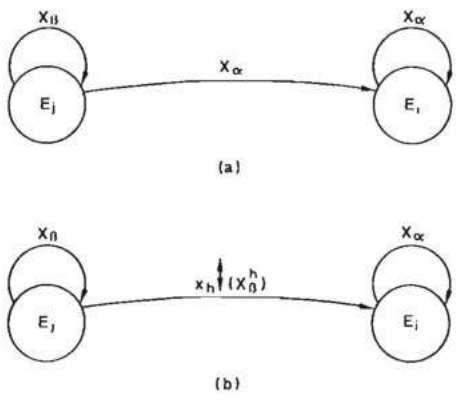


FIGURA 6.36.—**Representación gráfica de las especificaciones de cambio de estado interno de un sistema secuencial asíncrono caracterizado por niveles.**

Si se supone que es x_h la variable que cambia de estado al pasar del vector X_β al X_α, (fig. 6.49*b*) se puede definir X_α a partir de X_β por medio de la ecuación:

$$X_\alpha = x_h\updownarrow \cdot (X_\beta^h)$$

donde:

$x_h\updownarrow$: cambio de estado de la variable x_h. Si el cambio es de cero a uno se representa por $x_h \uparrow$ y si es de uno a cero por $x_h \downarrow$.

X_β^h: vector de entrada formado por todas las variables de entrada que coinciden con X_β excepto x_h.

·: producto lógico. Para simplificar puede suprimirse de forma explícita.

A continuación se aclaran los conceptos expuestos mediante un ejemplo. Supóngase un sistema secuencial asíncrono de tres variables de entrada x_1, x_2 y x_3, que permanece en el estado E_j estable mientras existe el vector de entrada $X_\beta = \bar{x}_1 x_2 \bar{x}_3$ y que debe cambiar de dicho estado E_j al E_i cuando el vector de entrada pase a ser $X_\alpha = \bar{x}_1 x_2 x_3$. El vector X_α se diferencia de X_β en el estado de x_3 y se puede definir:

$$X_\alpha = x_3 \uparrow (\bar{x}_1 x_2)$$

En este caso $x_h\updownarrow = x_3 \uparrow$ y $X_\beta^h = \bar{x}_1 x_2$.

La ecuación que se acaba de definir introduce una nueva operación lógica que es el producto del cambio de estado de una variable por un vector de todas las variables de entrada excepto la que cambia de estado. Esta ecuación constituye una nueva forma de representar las condiciones de transición entre los estados internos de un autómata asíncrono.

En el método de niveles, dado un estado E_j estable para un vector X_β, se analizan las n transiciones posibles representadas en la figura 6.37, lo cual resulta muy difícil de realizar cuando el número de variables de entrada es superior a tres.

Para reducir el número de transiciones a analizar a partir de un cierto estado interno se introduce el concepto de capacidad de transición C_T. Se define la capacidad de transición a partir de un cierto estado como una variable binaria que adopta el estado uno solamente cuando el cambio de estado de las variables de entrada producido a partir de un cierto estado interno debe hacer que el autómata cambie de estado. De la figura 6.37 se deduce la expresión general de la capacidad de transición a partir del estado E_j estable para el vector de entrada X_β:

$$C_T = x_1 \updownarrow (X_\beta^1) + x_2 \updownarrow (X_\beta^2) + \ldots + x_n \updownarrow (X_\beta^n)$$

que se puede abreviar:

$$C_T = \sum_{h=1}^{h=n} x_h \updownarrow (X_\beta^h)$$

donde Σ = suma lógica.

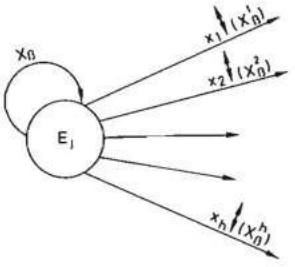


FIGURA 6.37.—**Representación gráfica de las transiciones posibles de un sistema secuencial asíncrono caracterizado por niveles a partir de un estado interno.**

Pero en muchas ocasiones no todos los posibles cambios de las variables de entrada han de ser memorizados mediante un cambio de estado interno y no necesitan por ello ser considerados. En este caso se tiene una expresión particular de la capacidad de transición:

$$C_{T1} = \Sigma\, x_h \updownarrow (X_\beta^h)$$

en la que la suma lógica no se extiende a todos los valores posibles de h.

Pero además, en ocasiones, a partir de un cierto estado interno se debe memorizar el cambio de estado de una variable de entrada x_h si dicho cambio se produce mientras un cierto número de variables, menor que $n - 1$, permanece en un cierto estado. Se obtiene de esta forma otra expresión particular de la capacidad de transición:

$$C_{T2} = \Sigma\, x_h \updownarrow (X'_\alpha)$$

donde X'_α representa un vector de entrada formado por un cierto número n' de variables de entrada $[n' < (n - 1)]$.

Todavía existe una expresión particular más sencilla de la capacidad de

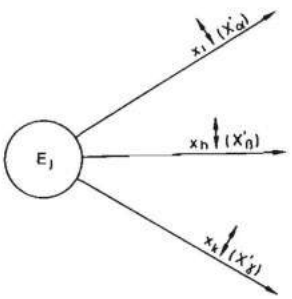


FIGURA 6.38.—**Representación gráfica de la expresión de la capacidad de transición $C_{T2} = \Sigma\, x_h\,(x'\ \alpha)$.**

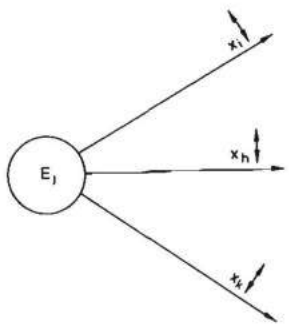


FIGURA 6.39.—**Representación gráfica de la expresión de la capacidad de transición $C_{T3} = \Sigma\, x_h$.**

transición, aquella que indica que el cambio de estado interno debe producirse cuando cambia de estado una variable de entrada independientemente del estado de las demás variables de entrada en ese instante. Esta expresión es:

$$C_{T3} = \Sigma\, x_h\updownarrow$$

En las dos expresiones particulares de la capacidad de transición C_{T2} y C_{T3}, representadas gráficamente en las figuras 6.38 y 6.39, desaparece el concepto de estabilidad del estado interno para un determinado vector de entrada. Se considera que un estado interno es estable mientras la capacidad de transición a partir de él no adopte el estado lógico uno. Por ello se define realmente mediante las ecuaciones indicadas en párrafos anteriores un nuevo concepto de estado interno, estable, mientras la capacidad de transición a partir de él es cero, que se denominará estado interno de transición. En las figuras 6.40 y 6.41 se representan gráficamente los cambios de estado interno para las dos últimas expresiones de la capacidad de transición y en ellas se observa que no se especifica ninguna condición de permanencia en un cierto estado.

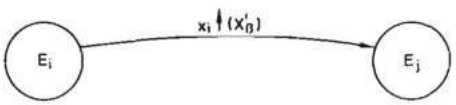


FIGURA 6.40.—**Representación gráfica de la transición entre estados internos cuando es igual a uno la expresión $x_i\uparrow\ (X'_\beta)$**

FIGURA 6.41.—**Representación gráfica de la transición entre estados internos cuando es igual a uno la expresión $x_i\uparrow$.**

Las expresiones algebraicas introducidas en párrafos anteriores permiten representar las condiciones de cambio de estado interno mediante los cambios de estado de las variables de entrada.

Pero en ocasiones debe producirse el cambio de estado interno si un cierto número de variables de entrada alcanza simultáneamente un cierto estado, lo cual es equivalente a la aparición de un determinado vector de entrada. Continuando en el supuesto de que una sola variable de entrada cambie de

estado simultáneamente, un vector de entrada X_α se puede alcanzar mediante el cambio de alguna de las n variables de entrada a partir de un vector que se diferencie de X_α en el estado de una única variable.

Por ello X_α se puede representar mediante la expresión algebraica:

$$X_\alpha = x_1 \updownarrow (X_\alpha^1) + x_2 \updownarrow (X_\alpha^2) + \ldots + x_i \updownarrow (X_\alpha^i) + \ldots + x_n \updownarrow (X_\alpha^n)$$

y el cambio de estado interno en estas circunstancias se representa tal como se indica en la figura 6.42.

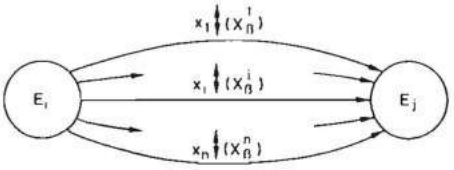


FIGURA 6.42.—**Representación gráfica de la transición entre estados internos cuando se alcanza un determinado vector de entrada X_α.**

Pero las n transiciones indicadas en la figura 6.42 se pueden reducir a una sola si se introduce el concepto de cambio de estado de un vector de entrada X_α. En la figura 6.43 se representa este concepto de forma gráfica. El autómata ha de pasar al estado E_j si se encuentra en el estado E_i y se produce una transición del estado cero al uno del vector X_α, lo cual se representa mediante el símbolo $X_\alpha \uparrow$.

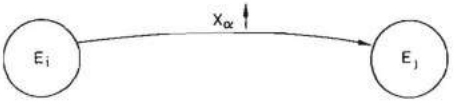


FIGURA 6.43.—**Representación gráfica de la transición entre estados internos cuando pasa de cero a uno el vector X_α.**

La expresión de la capacidad de transición a partir de un cierto estado E_i puede estar formada por una combinación de las diferentes expresiones estudiadas en los apartados anteriores, tal como se indica en la figura 6.44.

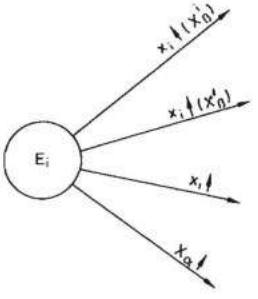


FIGURA 6.44.—**Representación gráfica de una capacidad de transición.**

6.2.3.2 Diagrama de secuencia de operaciones y grafo de transición de un autómata asíncrono de control. En el apartado anterior se expusieron los conceptos de cambios de estado interno de un autómata asíncrono de control en función de los cambios de estado de las variables de entrada o de los vectores de entrada. La principal ventaja del método de síntesis que se está analizando es la obtención directa de un grafo mínimo de evolución del autómata a partir de las especificaciones de funcionamiento.

Para obtener este grafo de evolución o de transiciones del autómata a partir de las especificaciones de funcionamiento es conveniente representar éstas de forma gráfica, lo cual se realiza mediante el diagrama de secuencia de operaciones. Este diagrama ha de cumplir dos condiciones muy importantes:

a) Obtenerse fácilmente a partir de las especificaciones de funcionamiento sin necesidad de tener conocimientos de álgebra binaria.

b) Estar estructurado de tal forma que a partir de él se obtenga de forma directa un grafo de evolución del autómata que permita su síntesis sistemática aplicando los conceptos teóricos desarrollados en el apartado anterior.

El diagrama de secuencia de operaciones de un autómata asíncrono de control debe expresar la secuencia de activación y desactivación de las variables de salida cuando las variables de entrada evolucionan de acuerdo con las especificaciones de funcionamiento del sistema.

Para poder representar mediante un diagrama de secuencia las especificaciones de funcionamiento, es necesario crear símbolos adecuados para cada una de las diferentes acciones de un autómata asíncrono de control, que son:

a) Tomas de decisión de cambio de estado interno en función del valor de la capacidad de transición C_T a partir de cada estado interno. Se representan mediante un rombo en cuyo interior se indica la expresión de la capacidad de transición.

b) Activación o desactivación de las variables de salida representadas mediante un paralelogramo en cuyo interior se indica la variable o variables que se activan o desactivan.

Ambos símbolos se representan en la figura 6.45.

Las especificaciones de funcionamiento del autómata se representan de forma gráfica mediante los símbolos citados adecuadamente enlazados entre sí, constituyendo el diagrama de secuencia de operaciones.

En este diagrama está contenida de una forma implícita la evolución del estado interno de transición del autómata. Para hacer que esta información se haga explícita solamente es necesario especificar las acciones de memorización que deberá realizar el autómata después de que en una toma de decisión la capacidad de transición adopte el estado lógico uno. Estas acciones de memorización se convierten en cambios de estado interno. Por ello en el diagrama de secuencia de operaciones se pueden representar mediante un círculo en cuyo interior se indica el estado al cual ha de llegar el autómata cuando la capacidad de transición C_T toma el valor lógico uno.

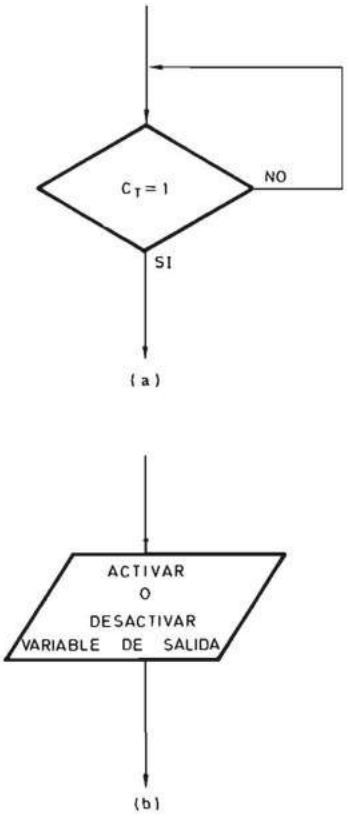


FIGURA 6.45.—**Símbolos de las acciones de un autómata asíncrono de control: *a*) Toma de decisión de cambio de estado interno; *b*) Activación o desactivación de una variable de salida.**

Mediante un par de ejemplos se ilustrará la realización del diagrama de secuencia de operaciones y la obtención del grafo de transiciones a partir de él.

Ejemplo 6.2:

Una vía férrea con tráfico en ambos sentidos corta a una carretera en la cual se coloca una barrera gobernada por la salida Z de un autómata asíncrono. A 500 m del punto de cruce se colocan dos detectores x_1 y x_2 respectivamente. A partir de un estado inicial en el que $z = 0$, ésta debe pasar al estado uno cuando se acerca un tren en cualquier sentido al rebasar su máquina los 500 m del cruce y debe volver al estado cero cuando el último vagón se aleja más de dicha distancia independientemente de la longitud del tren.

En la figura 6.46 se representa el conjunto del sistema. Se obtendrá en primer lugar el diagrama de secuencia de operaciones.

Al existir una vía única, no es posible que circulen dos trenes en sentidos opuestos simultáneamente.

En el instante de dar tensión, el autómata asíncrono deberá ponerse en estado inicial, en el cual la variable de salida $Z = 0$. A partir de este instante el autómata debe observar si se produce una conmutación de x_2 o de x_1 del estado cero al uno, lo cual indica que llega al cruce un tren en el sentido de izquierda a derecha o al contrario, respectivamente. Si es x_2 la que pasa al estado uno, debe activarse la salida Z y permanecer activada hasta que el tren abandone la zona compredida entre ambos detectores, lo cual se detecta mediante

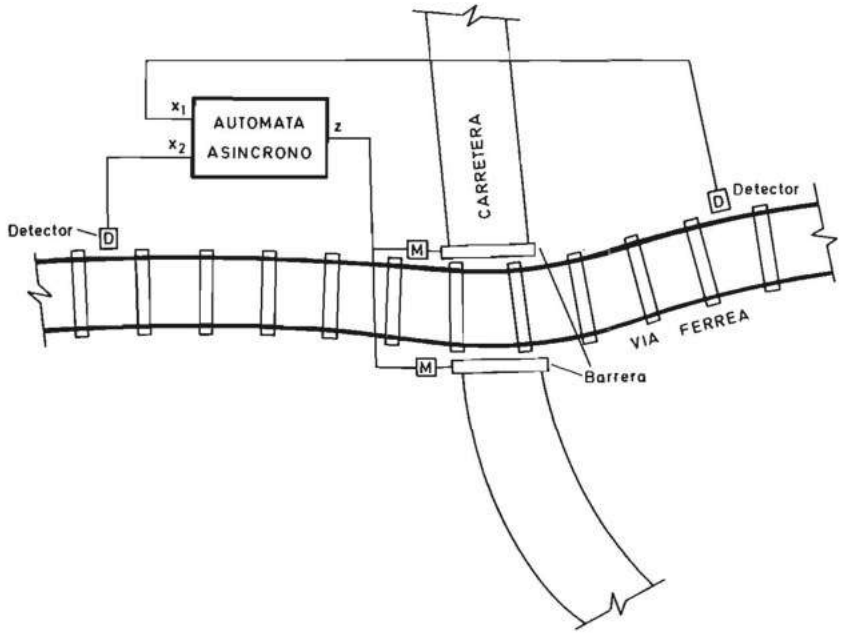


FIGURA 6.46.—**Barrera de un paso a nivel controlada por un autómata asíncrono.**

la conmutación de uno a cero de x_1. Si, por lo contrario, es x_1 la que pasa primero de uno a cero, lo cual quiere decir que el tren va de derecha a izquierda, debe activarse también la variable de salida Z y permanecer activada hasta que la variable x_2 pase de uno a cero, lo cual indica que el tren ha abandonado la zona comprendida entre ambos detectores. En los dos casos anteriores, el autómata debe volver al estado inicial para esperar la llegada de un nuevo tren.

Las especificaciones que se acaban de describir se expresan gráficamente mediante el diagrama de secuencia de operaciones de la figura 6.47 que se deduce a partir de aquéllas de forma inmediata.

En la figura 6.48 se representa el diagrama de secuencia de operaciones, indicándose en él los estados que alcanza el autómata después de cada toma de decisión. El estado inicial se denomina E_1. A partir de él se pasa a E_2 si se produce $x_1 \uparrow$, o a E_3 si se produce $x_2 \uparrow$. E_2 memoriza que ha llegado un tren de derecha a izquierda y E_3 que lo ha hecho en sentido contrario. De E_2 se debe pasar a E_1

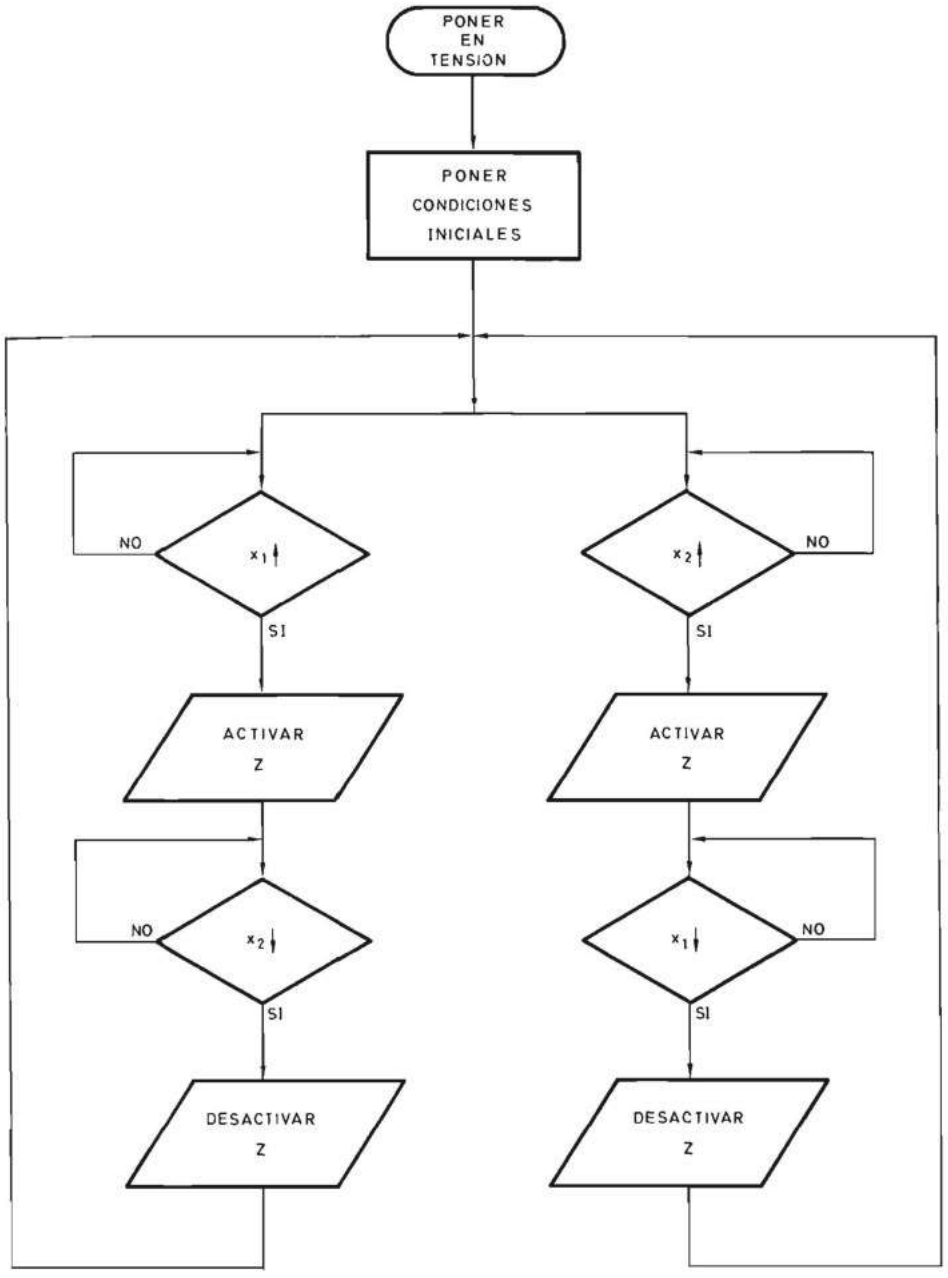


FIGURA 6.47.—**Diagrama de secuencia de operaciones del autómata asíncrono del ejemplo 6.2.**

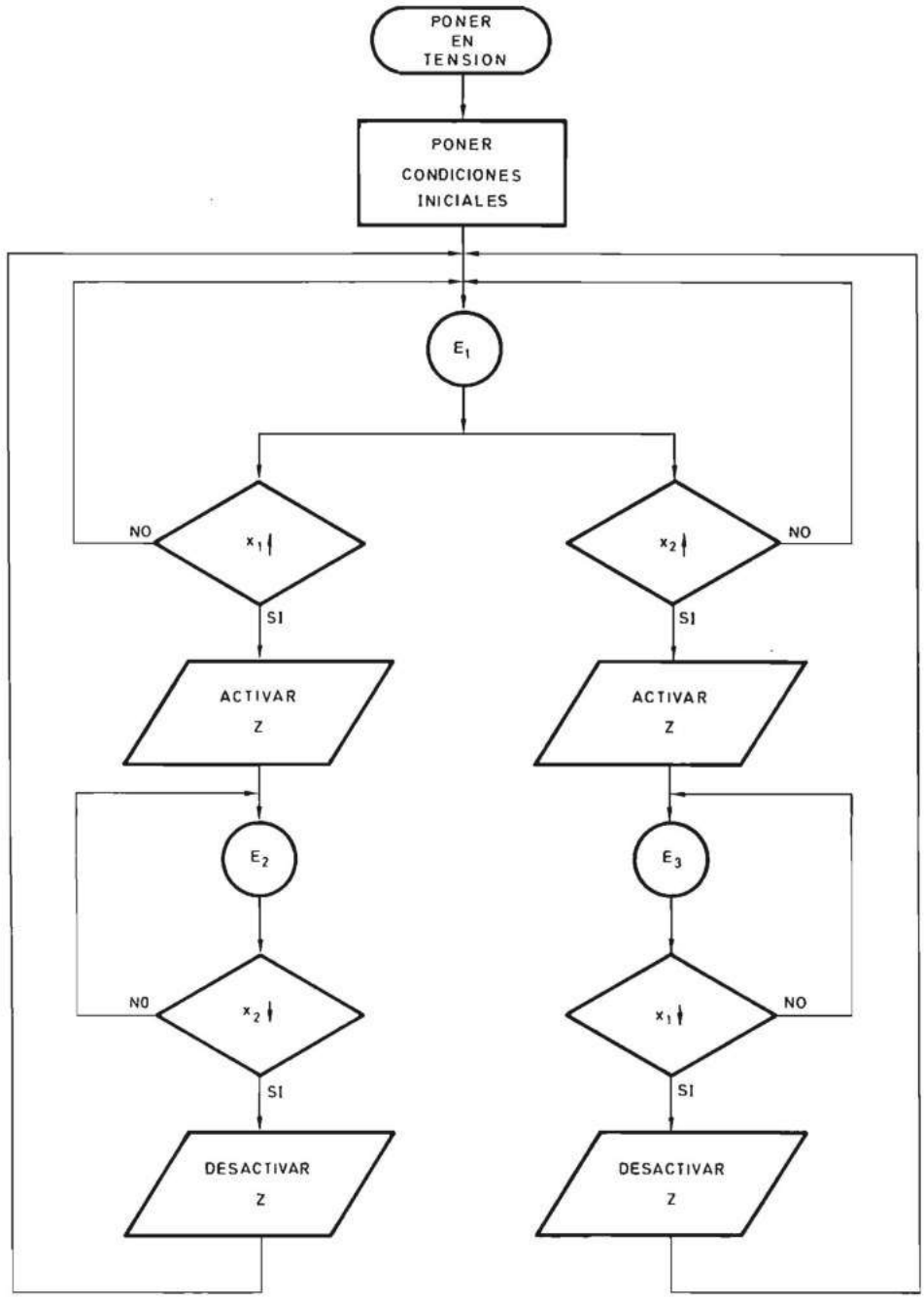


FIGURA 6.48.—**Diagrama de secuencia de operaciones del autómata asíncrono del ejemplo 6.2 en el que se incluyen los estados internos.**

al conmutar $x_2 \downarrow$ y, por el contrario, cuando x_2 pasa de cero a uno ($x_2\uparrow$) el autómata debe permanecer en E_2. Por ello la capacidad de transición a partir de E_2 es $C_T = x_2 \downarrow$. De igual forma se deduce que de E_3 sólo se debe pasar a E_1 si se produce $x_1 \downarrow$ y la capacidad de transición a partir de E_3 es $C_T = x_1 \downarrow$.

A partir de la figura 6.48 se obtiene de forma inmediata el grafo de transición. Para ello se asignan puntos de un plano a cada estado y se enlazan éstos por medio de segmentos dirigidos sobre los cuales se indica la expresión algebraica cuya igualdad a uno produce el cambio de estado. En cada uno de los estados se indican las variables de salida que adoptan el valor uno. En este caso existe una única variable de salida Z que toma el valor uno en los estados E_2 y E_3. El grafo de transiciones obtenido es el representado en la figura 6.49.

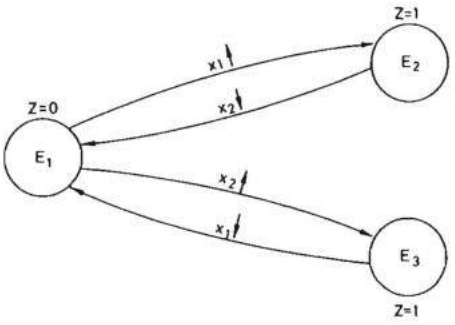


FIGURA 6.49.—**Grafo de transiciones del autómata asíncrono del ejemplo 6.2.**

Ejemplo 6.3:

Un autómata asíncrono de control tiene como variables de entrada x_1 y x_2 y genera la variable de salida z que inhibe la puesta en marcha de la máquina M. En el instante de dar tensión a la máquina y al autómata, lo cual se realiza simultáneamente, éste debe quedar en un estado inicial en el que la variable de salida Z se encuentra en estado cero e inhibe la puesta en marcha de M. La salida Z debe pasar al estado uno y desinhibir la actuación de la máquina M si la variable x_1 pasa del estado cero al uno mientras x_2 se encuentra en estado cero y seguidamente se produce el cambio de cero a uno de x_2 estando x_1 en estado cero. Entre los dos cambios especificados x_1 no debe cambiar de cero a uno. En la figura 6.50 se representa el diagrama de bloques de la máquina M y el autómata que la controla.

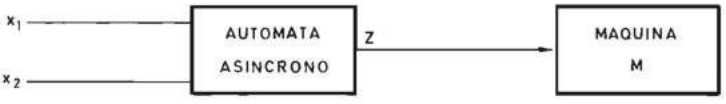


FIGURA 6.50.—**Diagrama de bloques del sistema del ejemplo 6.3.**

En este ejemplo se comprobará que aparecen las expresiones de la capacidad de transición iguales al producto del cambio de una variable por un vector de las variables de entrada.

En la figura 6.51 se representa el diagrama de secuencia de operaciones. En el instante de dar tensión el autómata recibe una puesta en condiciones iniciales. A partir de dicho instante el autómata observa si se produce una transición de x_1 y simultáneamente la variable x_2 se encuentra en estado cero, es decir, si se produce la igualdad a uno de la capacidad de transición $C_T = x_1 \uparrow (\bar{x}_2)$. Mientras no se produce dicha igualdad, el autómata permanece en el estado en que se encontraba, y cuando la detecta, la memoriza y por lo tanto pasa a un nuevo estado. Si a continuación se produce un cambio

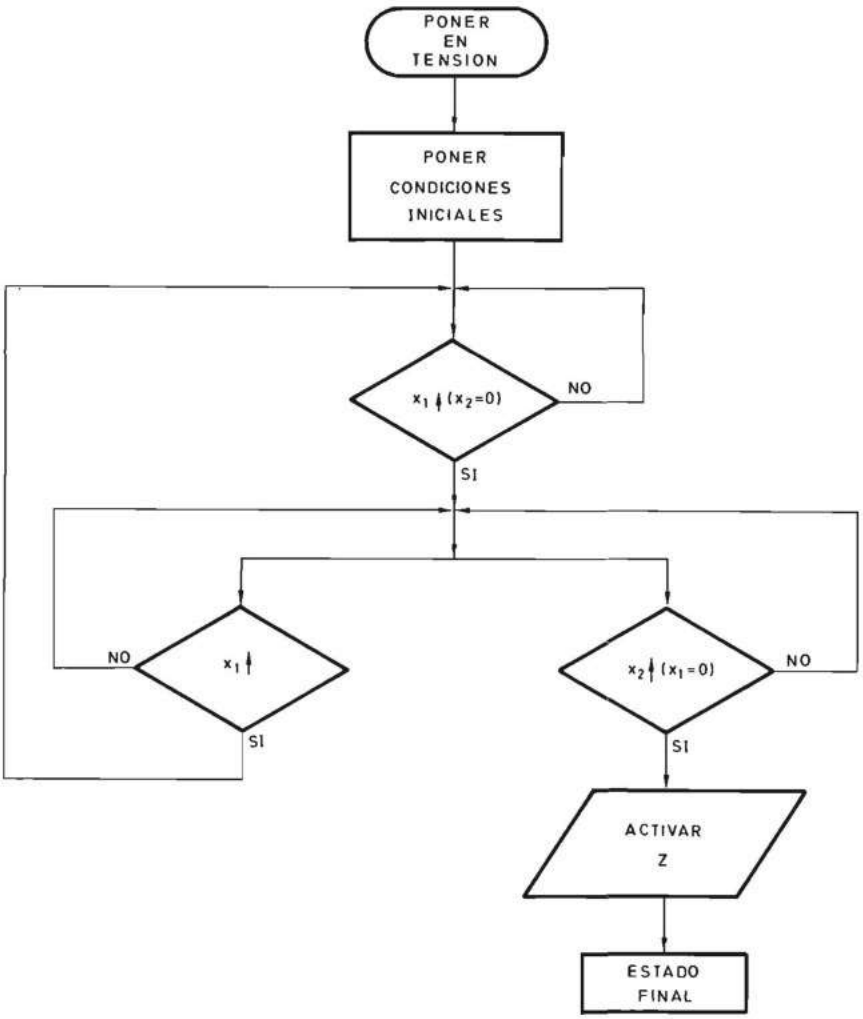


FIGURA 6.51.—**Diagrama de secuencia de operaciones del autómata asíncrono del ejemplo 6.3.**

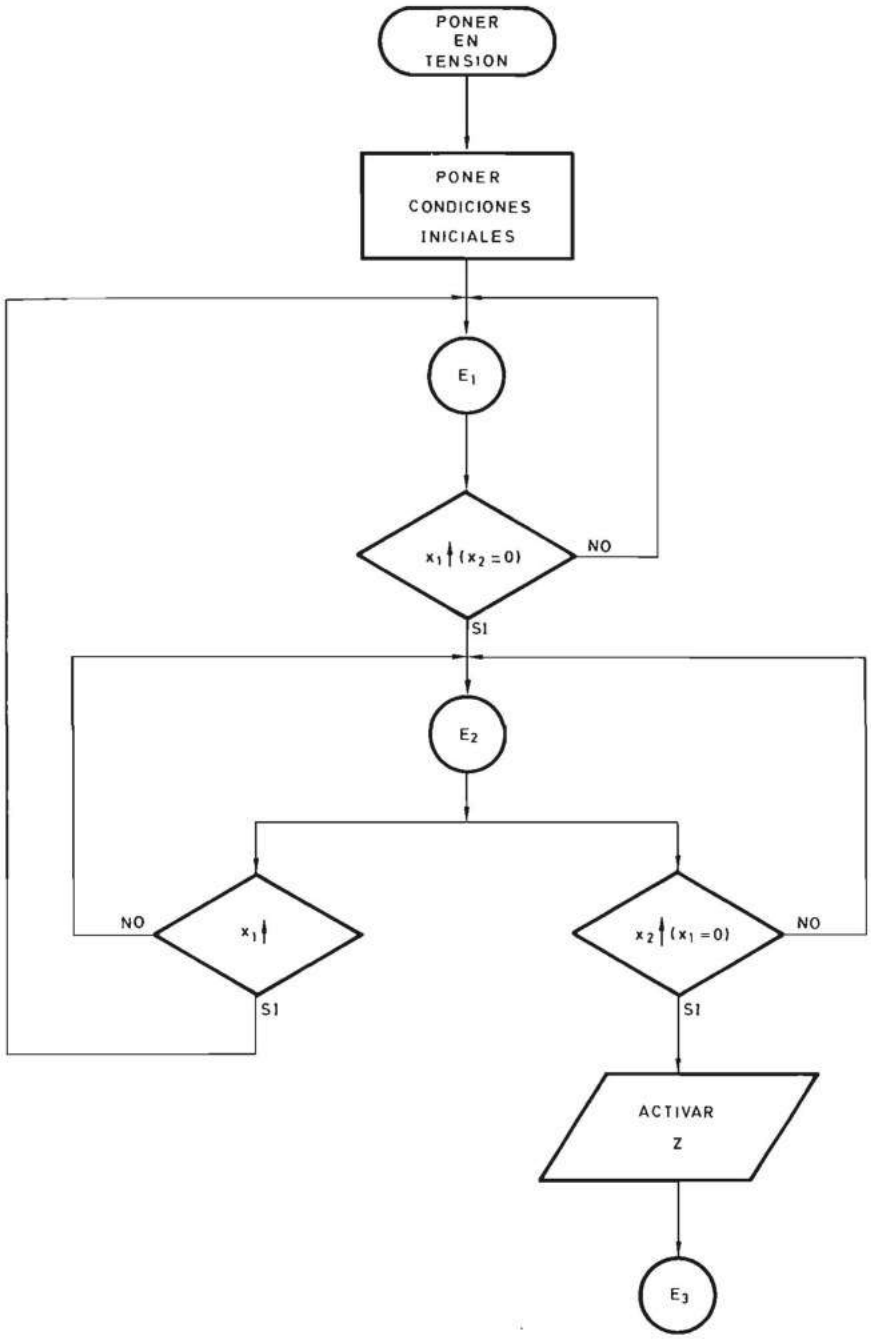


FIGURA 6.52.—**Diagrama de secuencia de operaciones del autómata asíncrono del ejemplo 6.3, en el que se incluyen los estados internos.**

de cero a uno de la variable x_1, el autómata vuelve al estado inicial porque no se ha producido la secuencia correcta; si, por el contrario, es la variable x_2 la que cambia de cero a uno mientras la variable x_1 se encuentra en estado cero, se produce la secuencia correcta, se activa la salida Z y el autómata alcanza un estado del cual nunca sale porque su capacidad de transición es siempre nula ($C_T = 0$). De esta forma la máquina M queda desinhibida y puede iniciar su funcionamiento.

En la figura 6.52 se representa el diagrama de secuencia de operaciones con los estados de transición indicados de forma explícita. Utilizando este diagrama el lector puede deducir el grafo de transiciones del autómata representado en la figura 6.53.

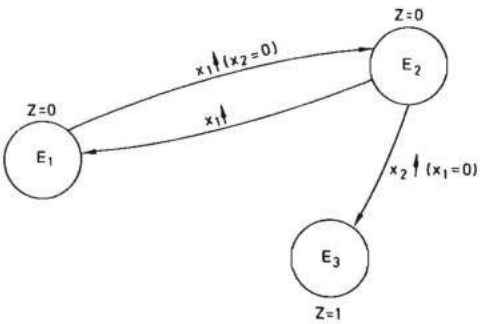


FIGURA 6.53.—**Grafo de transiciones del autómata asíncrono del ejemplo 6.3.**

6.2.3.3 Biestables asíncronos accionados por los cambios de nivel. Son sistemas secuenciales asíncronos con una sola variable de salida en los que las variables de entrada actúan por medio de los cambios de nivel.

Existen los mismos tipos estudiados en el apartado 6.2.2.2 que se suelen denominar con letras minúsculas para distinguirlos de aquellos *s-r, j-k, p-q*.

En la figura 6.54 se indica como ejemplo el diagrama de secuencia de activación y desactivación de la variable de salida Z de un biestable *s-r* en el cual ambas

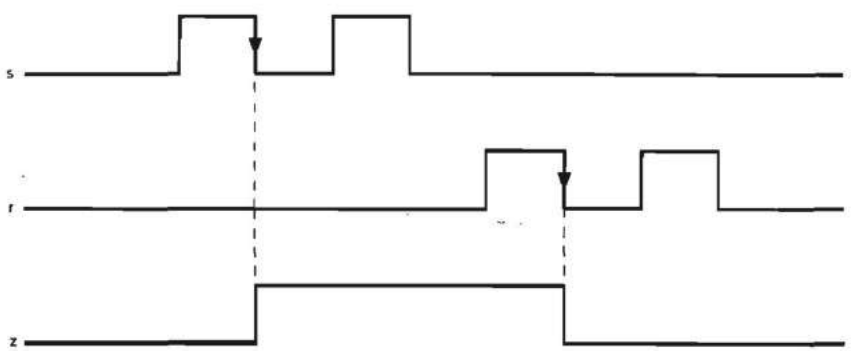


FIGURA 6.54.—**Diagrama de secuencia temporal que relaciona las variables de entrada (*s* y *r*) y la variable de salida (*z*) del biestable *s-r* activado por cambio del nivel uno al cero.**

entradas actúan sobre el biestable en el instante en que cambian del estado uno al cero. Cuando *s* conmuta del estado uno al cero, la salida *Z* pasa al estado uno y cuando es *r* la que realiza dicha conmutación, *Z* pasa al estado cero. Se supone que *s* y *r* no pueden encontrarse nunca en estado uno simultáneamente. El lector puede deducir el diagrama de secuencia de operaciones de la figura 6.55 que representa también la forma de actuación del biestable *s-r* activado por los cambios de cero a uno de las variables de entrada.

Se define a continuación un biestable asíncrono que combina los niveles y cambios de nivel y que será utilizado como célula básica para la síntesis de autómatas asíncronos y por ello se le denomina célula asíncrona activada por flancos (CAF).

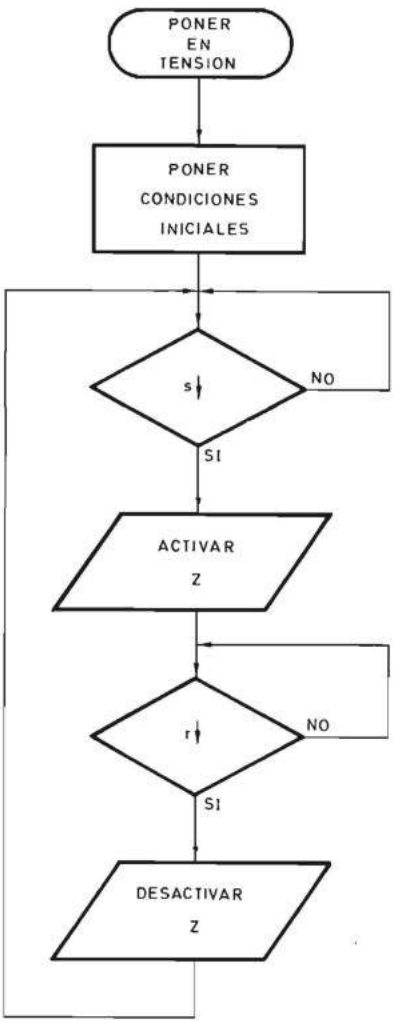


FIGURA 6.55.—**Diagrama de secuencia de operaciones del biestable *s-r*.**

Esta célula posee una variable de salida Q y tres variables de entrada D, T y T'. En la figura 6.56 se representa el diagrama de bloques y el diagrama de transiciones de CAF. La célula pasa del estado cero al uno, si se produce un cambio del estado cero al uno de la entrada T ($C1$), estando simultáneamente la entrada D ($1D$) en estado lógico uno. Esto se representa en la figura 6.56*b* mediante un segmento dirigido desde el estado $Q = 0$ al $Q = 1$ sobre el cual se indica la expresión $T \uparrow (D = 1)$.

El cambio de cero a uno de la variable T se representa mediante $T \uparrow$ y la simultaneidad de dicho cambio y el estado uno de D mediante la expresión $T \uparrow (D = 1)$.

La célula pasa del estado uno al cero, si se produce un cambio del estado cero al uno de la variable T' lo cual se representa en la figura 6.56*b* mediante un segmento dirigido desde el estado $Q = 1$ al $Q = 0$ sobre el cual se indica $T' \uparrow$. La célula asíncrona activada por flancos (CAF) tiene también entradas de puesta a cero y puesta a uno por nivel, S y R, respectivamente que permiten fijar el estado de la célula en el instante inicial de aplicar la tensión de alimentación. La forma de realizar físicamente la célula CAF utilizando bloques funcionales disponibles en circuito integrado en las diversas tecnologías se estudia en el apartado 6.3.3.

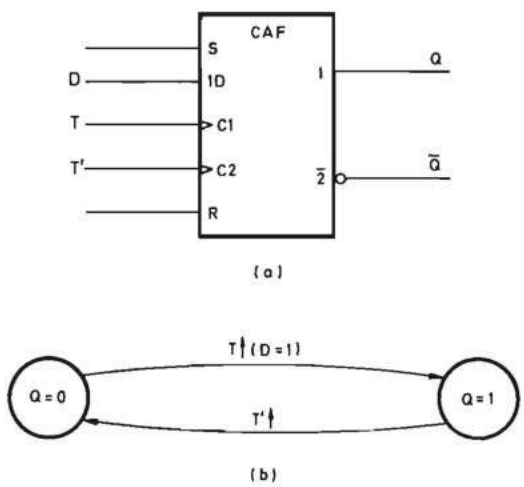


FIGURA 6.56.—**Célula activada por flancos (CAF): *a*) Símbolo lógico; *b*) Diagrama de transiciones.**

En la figura 6.56*a* se utiliza un símbolo normalizado para representar la célula CAF. La entrada T se denomina $C1$ y lleva incorporado el indicativo de flanco de subida. La entrada D se denomina $1D$ y en la salida Q se pone el símbolo 1. De esta forma se indica que Q pasa al estado uno cuando se aplica un flanco activo (de subida en este caso) en $C1$ y $1D$ se encuentra el nivel uno. La entrada T' se denomina $C2$ y lleva incorporado el indicativo del flanco de subida, y en la salida $\overline{Q}$ se pone el símbolo $\overline{2}$; de esta forma se indica que un flanco activo en $C2$ pone a uno $\overline{Q}$, es decir, Q en estado cero.

6.2.3.4 Síntesis de los sistemas secuenciales asíncronos con células asíncronas activadas por flancos (CAF). Seguidamente se desarrollará un sistema físico para realizar los diagramas de secuencia de operación utilizando como bloques fundamentales las células asíncronas activadas por flancos (CAF).

La asignación de células CAF a los estados internos del grafo de transición constituye la codificación. Existen diferentes formas de resolver este problema:

a) Tratar de reducir al mínimo el número n de células CAF necesarias, de tal forma que $2^n \geq m$, siendo m el número de estados de transición del grafo. Esta solución complica el sistema combinacional que enlaza entre sí las células CAF para constituir el autómata, e introduce el problema de la conmutación simultánea de varias variables de estado interno que ha de ser tenido en cuenta para evitar los cambios inadecuados entre estados.

b) Asignar una célula CAF a cada estado interno de transición de tal forma que dicha célula esté en estado uno solamente si el autómata se encuentra en el estado correspondiente. Esta solución eleva el número de células necesarias, pero reduce la complejidad del sistema combinacional y sobre todo evita sistemáticamente la aparición de fenómenos aleatorios tal como se demuestra en el apartado 6.2.3.5. Por ello se ha elegido esta solución.

Llegado este punto es conveniente adelantar que el progreso de la microelectrónica está cambiando los métodos prácticos de diseño de los sistemas digitales al reducir la influencia de la complejidad del sistema físico (hardware). De esta forma, los conceptos más importantes pasan a ser la sistematización del diseño mediante la adopción de técnicas de diseño asistido por computador, la modularidad y la programabilidad. Por ello, los sistemas secuenciales asíncronos están siendo sustituidos por sistemas secuenciales síncronos que reciben el nombre de controladores lógicos que se estudian más adelante en el apartado 6.3.5.6.3. De ahí que la síntesis con células CAF tenga cada día menor importancia.

No obstante a continuación se estudia el método de realización de autómatas asíncronos con células CAF.

Tal como se ha indicado, se asigna una célula a cada estado interno de transición del grafo de tal forma que dicha célula esté en estado uno solamente cuando el autómata se encuentre en el estado interno de transición correspondiente.

En primer lugar se deduce la forma de hacer pasar las células al estado uno, lo cual depende de la expresión de la capacidad de transición.

En la figura 6.57*a* se representa el cambio de estado interno para la expresión de $C_T = x_h \uparrow (X_\beta^h)$ y en la figura 6.57*b* el esquema correspondiente. La variable x_h se conecta a la entrada T ($C1$) de la célula CAF asignada al estado E_i. A la entrada D ($1D$) de esta célula se conecta el producto de Q_j (salida de la célula asignada a E_j) y las variables x_1 *a* x_n excepto x_h en forma directa o inversa según el vector de entrada X_β^h de que se trate (por ello se representan las variables x_1 a x_n con el superíndice *). Si el cambio de estado se ha de producir cuando cambia x_h, pero sólo un cierto número de las variables de entrada forman el vector X'_β, el esquema es idéntico con la excepción de que solamente las variables que forman parte de X'_β se conectarán a la puerta Y unida a la entrada D ($1D$) de la célula

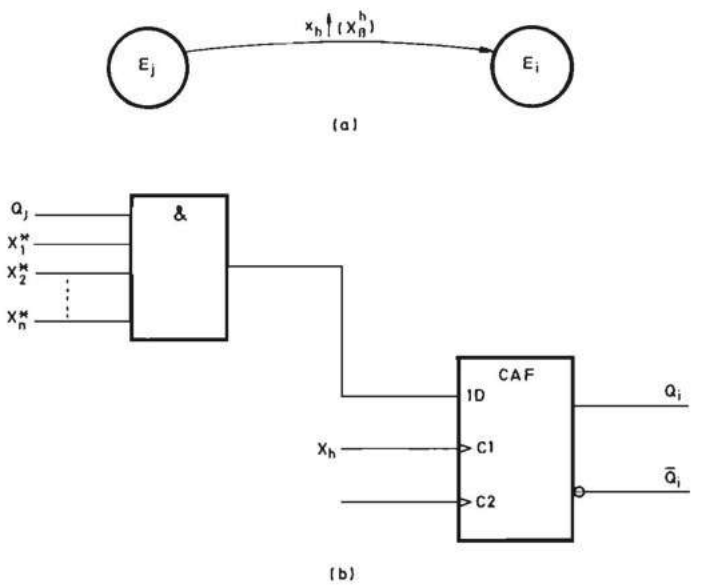


FIGURA 6.57.—**Cambio del estado interno, E_j al E_i cuando es igual a uno la expresión $x_h\uparrow$ (X_β^h): *a*) Representación gráfica; *b*) Sistema físico realizado con una célula CAF.**

Si el cambio de estado interno debe producirse cuando x_h pasa de uno a cero ($x_h \downarrow$), se conecta la variable $\bar{x}_h$ a la entrada T ($C1$) (se supone que la célula CAF pasa al estado uno cuando su entrada T ($C1$) (conmuta de cero a uno).

Si el cambio de estado interno debe producirse al conmutar de cero a uno una variable de entrada x_h independientemente del estado de las demás (fig. 6.58*a*), el esquema correspondiente es el de la figura 6.58*b* en la que la variable x_h se co-

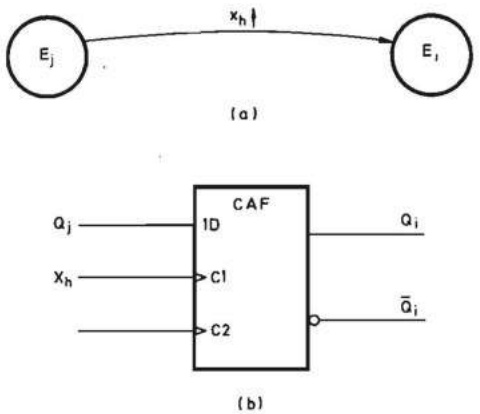


FIGURA 6.58.—**Cambio del estado interno E_j al E_i cuando $C_T = x_h\uparrow = 1$: *a*) Representación gráfica; *b*) Sistema físico realizado con una célula CAF.**

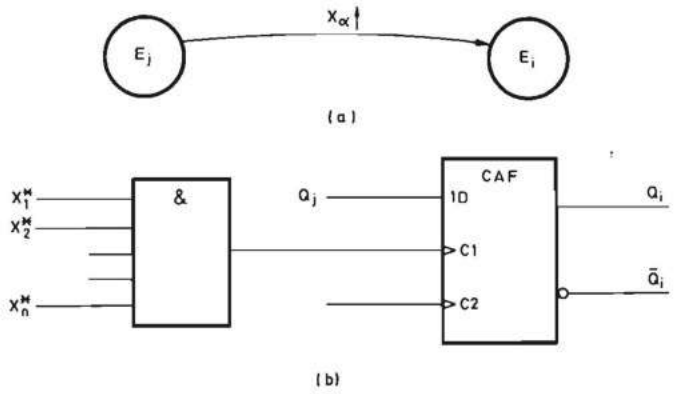


FIGURA 6.59.—**Cambio del estado E_j al E_i cuando es igual a uno la expresión $X_\alpha\uparrow$: *a*) Representación gráfica; *b*) Sistema físico realizado con una célula CAF.**

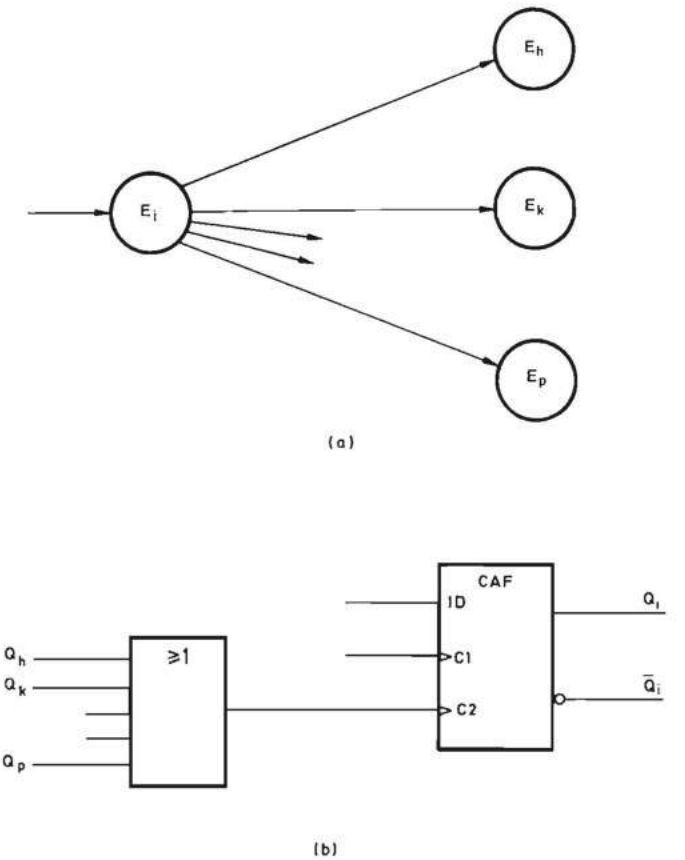


FIGURA 6.60.—**Borrado o desactivación de una célula: *a*) Representación gráfica; *b*) Sistema físico.**

necta a la entrada T ($C1$) y la salida Q_j de la célula asignada al estado E_j se conecta a la entrada D ($1D$).

Si el cambio de estado interno debe producirse al aparecer un determinado vector de entrada (fig. 6.59*a*), el esquema correspondiente es el representado en la figura 6.59*b*. La variable Q_j asignada al estado E_j se conecta a la entrada D ($1D$) y la entrada T ($C1$) se une a la salida de una puerta Y que realiza el producto de las variables de entrada que forman el vector de entrada cuya detección se quiere realizar.

A continuación se deducirá el circuito de borrado o desactivación de una célula. En general, a partir de un estado interno de transición E_i el autómata puede pasar a un cierto número de estados tal como se indica en la figura 6.60*a*. La célula Q_i debe borrarse cuando el autómata alcanza cualquiera de los estados sucesores de E_i, lo cual se consigue haciendo T'_i ($C2$) igual a una O lógica de las salidas Q de las células correspondientes (6.60*b*).

En los párrafos anteriores se ha supuesto que el estado E_i se alcanzaba a través de una sola transición. En ocasiones puede suceder, tal como se indica en la figura 6.61, que se llegue a E_i a partir de diferentes estados con expresiones distintas de la capacidad de transición. Pero la célula CAF sólo puede ser activada por una sola expresión de C_T. Para resolver este problema se crea una macrocélula, que se denomina célula asíncrona activada por flancos generalizada, cuyo diagrama de bloques y grafo de transición se representan en la figura 6.62. La célula CAF generalizada tiene una sola entrada T' ($C2$) pero posee varios pares de entrada D-T ($1D$-$C1$) que permiten su activación para otras tantas expresiones de la capacidad de transición a partir de estados diferentes.

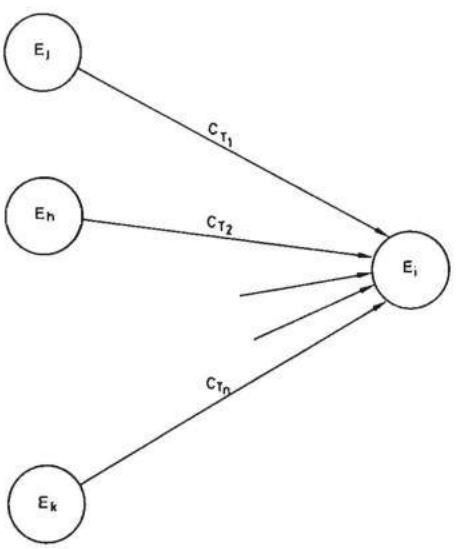


FIGURA 6.61.—**Representación gráfica de la activación de un estado a través de varias transiciones.**

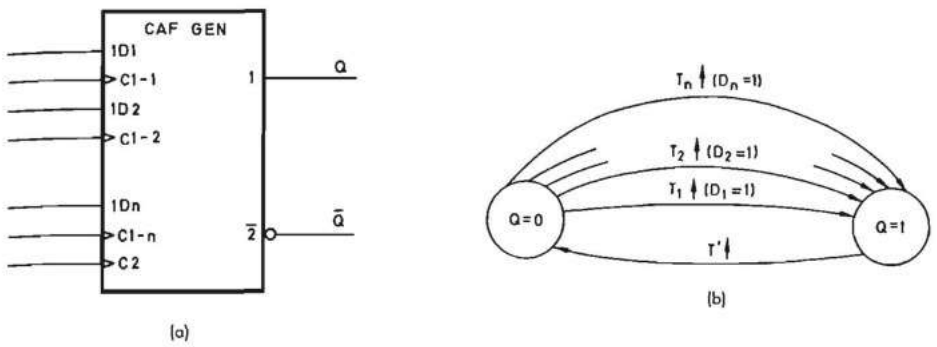


FIGURA 6.62.—**Célula CAF generalizada: *a*) Símbolo lógico; *b*) Grafo de transiciones.**

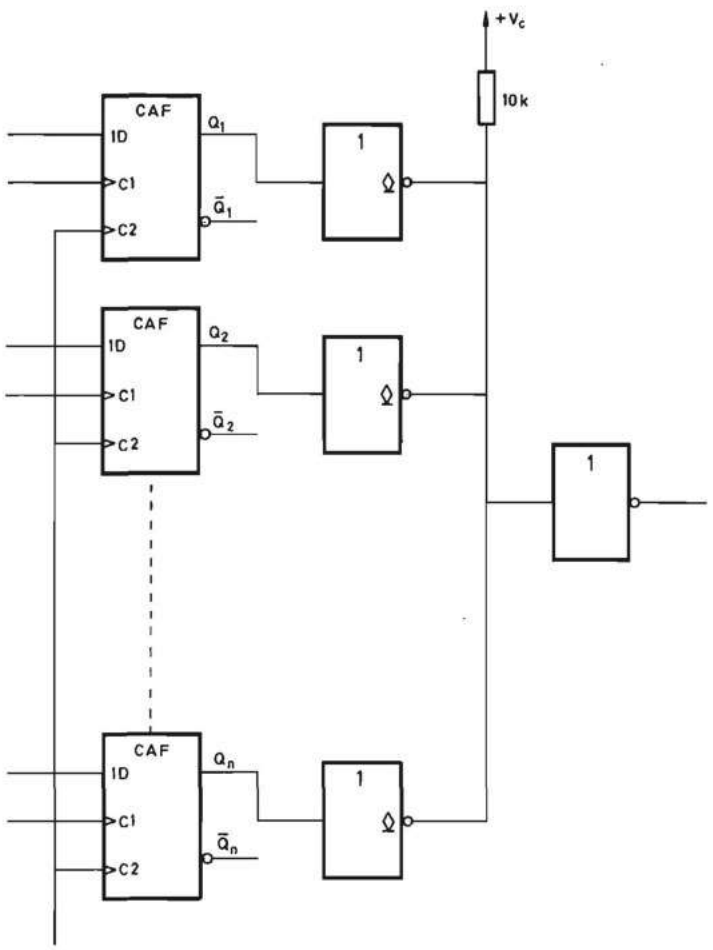


FIGURA 6.63.—**Célula CAF generalizada realizada con células CAF y puertas inversoras de colector abierto.**

La célula CAF generalizada se puede realizar de forma modular a partir de la célula CAF elemental representada en la figura 6.56. El esquema correspondiente se representa en la figura 6.63. Las entradas T' ($C2$) de todas las células CAF se unen entre sí para formar la entrada T' ($C2$) de la célula CAF generalizada. Las entradas D y T de cada CAF constituyen las entradas de la célula CAF generalizada. Las salidas Q de cada CAF se unen entre sí a través de seguidores o inversores de colector abierto o de tres estados.

En la figura 6.63 se utilizan inversores de colector abierto. Cada Q_i se conecta a la entrada de un inversor. Cuando una Q_i se encuentra en estado uno, la conexión común está a cero, y la salida Q de la célula en uno. Cuando todas las Q_i se encuentran en cero, la conexión común está en nivel uno y la salida Q en cero. En la figura 6.64 se representa la solución con inversores de tres estados. Cada Q_i se conecta a la entrada de un inversor y a su desinhibición (EN); de esta forma cuando Q_i se encuentra en estado cero, la salida del inversor correspondiente se encuentra en tercer estado y cuando Q_i sube al estado uno, se desinhibe el inversor y su salida pasa al estado cero. Dado que solamente una Q_i puede estar en estado uno en cada instante porque sólo una de las expresiones de C_T realizada por una célula puede alcanzar dicho estado en un instante determinado, las salidas de los inversores se pueden unir entre sí y esta conexión común invertida constituye la salida Q de la célula CAF generalizada. Si no se dispone de inversores o seguido-

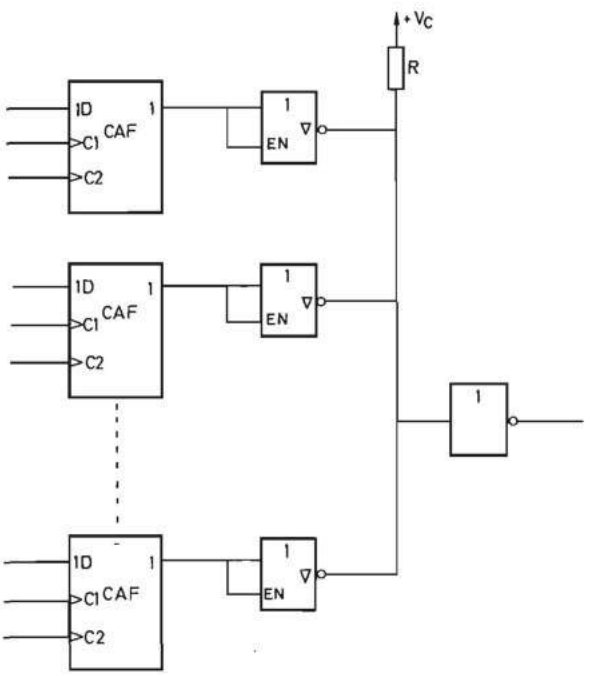


FIGURA 6.64.—**Célula CAF generalizada realizada con células CAF y puertas inversoras de tres estados.**

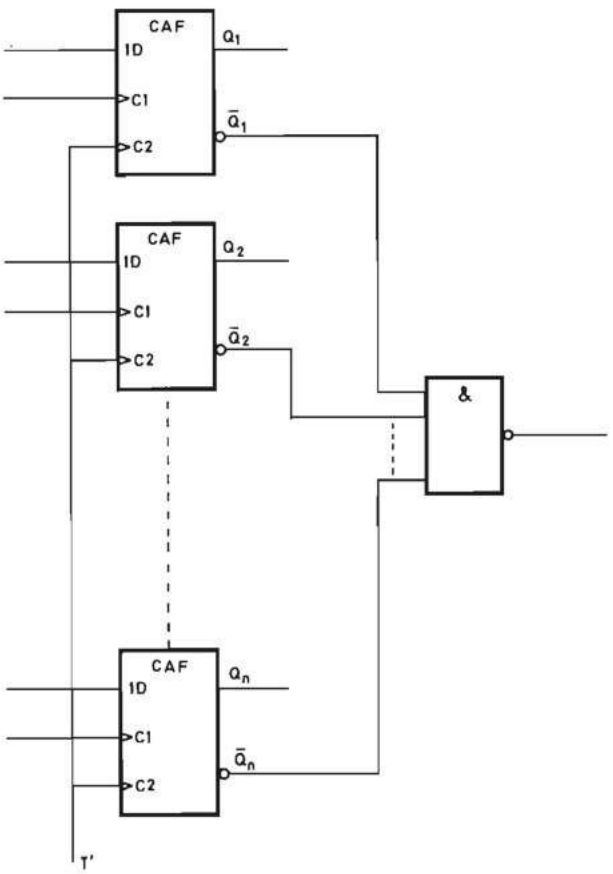


FIGURA 6.65.—**Célula CAF generalizada realizada con células CAF y una puerta NO-Y (NAND).**

res de tres estados, se pueden unir las salidas $\overline{Q}$ de las células CAF a las entradas de una puerta NO-Y (NAND) cuya salida constituirá la salida Q de la célula CAF generalizada. El inconveniente de esta solución es su falta de modularidad, porque el número de entradas de la puerta NO-Y se incrementa al hacerlo el número de células CAF que constituyen la célula CAF generalizada (fig. 6.65).

En la figura 6.66*a* se representa una parte de un grafo de transiciones que se utilizará como ejemplo de asignación de un estado E_i a una célula CAF generalizada. Se observa que a E_i se puede llegar desde cualquiera de los tres estados E_h, E_j o E_k y por ello se asigna a E_i una célula CAF generalizada formada por tres células CAF elementales. Se deducen seguidamente las ecuaciones de las entradas D ($1D$) y T ($C1$) de la célula CAF generalizada. Para ello se observan las expresiones de la capacidad de transición entre los estados correspondientes. Por ejemplo, para que el autómata pase de E_h a E_i es necesario que adopte el valor uno la capacidad de transición $C_T = x_i \uparrow \ (X^i_a)$ y por ello:

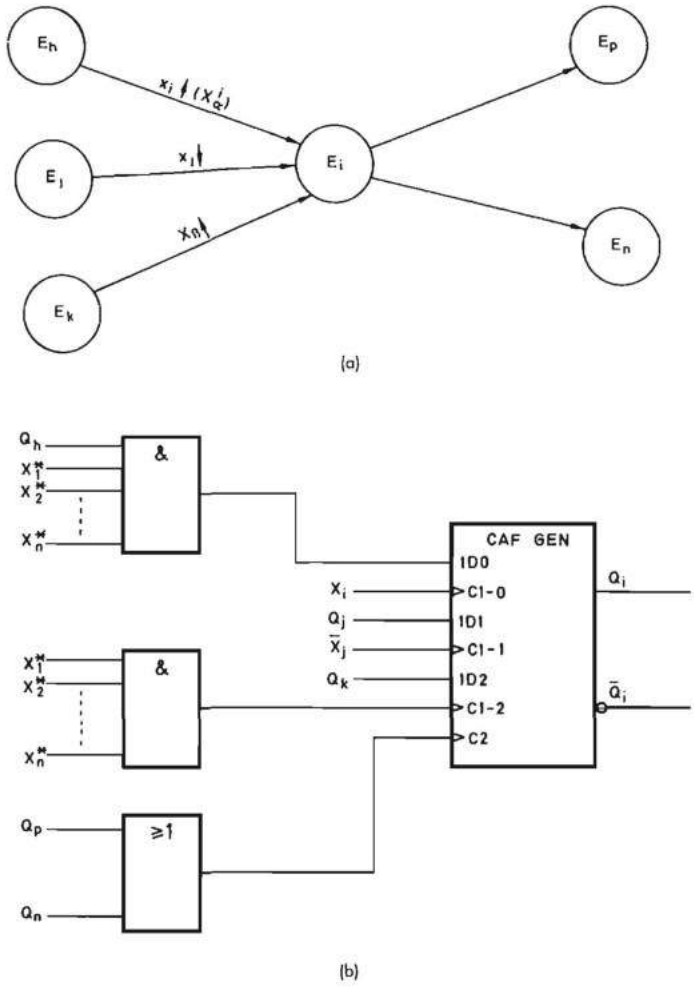


FIGURA 6.66.—**Ejemplo de asignación de una célula CAF generalizada a un grafo de transiciones: *a*) Grafo de transiciones; *b*) Esquema del sistema físico.**

$$D_{i-0}\ (1D0) = Q_h\, x_1^*\, x_2^* \ldots x_n^*$$
$$T_{i-0}\ (C1{-}0) = x_i$$

Basándose en lo explicado anteriormente (fig. 6.57 a 6.59) el lector puede deducir:

$$D_{i-1}\ (1D1) = Q_j$$
$$T_{i-1}\ (C1{-}1) = \bar{x}_j$$
$$D_{i-2}\ (1D2) = Q_k$$
$$T_{i-2}\ (C1{-}2) = x_1^*\, x_2^* \ldots x_n^*$$

En la figura 6.66*a* se observa que, a partir de E_i, el autómata puede ir a uno de los estados E_p, o E_n. Por ello la ecuación de la entrada T' $(C2)$ de la célula asignada a E_i resulta:

$$T'_i\ (C2) = Q_p + Q_n$$

A continuación, y como ejemplo de todo lo que se acaba de exponer, se deducen los esquemas de realización física con células CAF de los ejemplos 6.2 y 6.3 cuyos grafos de transición se representaron en las figuras 6.49 y 6.53 respectivamente.

Ejemplo 6.2:

En la figura 6.67 se repite el grafo de transición obtenido. Se asigna una célula CAF a cada estado interno de transición del grafo y se deducen las ecuaciones lógicas de las entradas de cada célula.

Al estado E_1 se puede llegar desde E_2 con una capacidad de transición $C_T = x_2 \downarrow$ o desde E_3 con $C_T = x_1 \downarrow$.

Por lo tanto, Q_1 es una célula CAF generalizada formada por dos células CAF elementales y las ecuaciones de sus entradas son:

$$D_{1-0}\ (1D1) = Q_3;\ T_{1-0}\ (C1—1) = \bar{x}_1$$
$$D_{1-1}\ (1D2) = Q_2;\ T_{1-1}\ (C1—2) = \bar{x}_2$$

A partir de E_1 se puede ir a E_2 o E_3 y por lo tanto:

$$T'_1\ (C2) = Q_2 + Q_3$$

Al estado E_2 sólo se puede llegar a partir de E_1 con $C_T = x_1 \uparrow$ y por lo tanto:

$$D_2\ (1D) = Q_1;\ T_2\ (C1) = x_1$$

De E_2 sólo se puede ir a E_1 y por lo tanto:

$$T'_2\ (C2) = Q_1$$

De igual forma el lector puede deducir para la célula Q_3:

$$D_3\ (1D) = Q_1;\ T_3\ (C1) = x_2$$
$$T'_3\ (C2) = Q_1$$

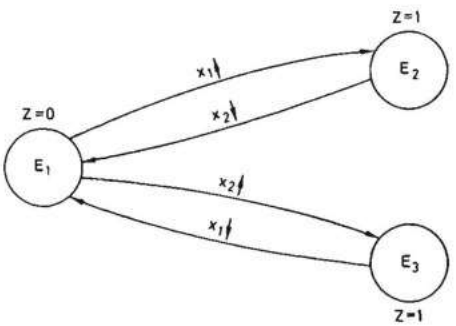


FIGURA 6.67.—**Grafo de transiciones del autómata asíncrono del ejemplo 6.2.**

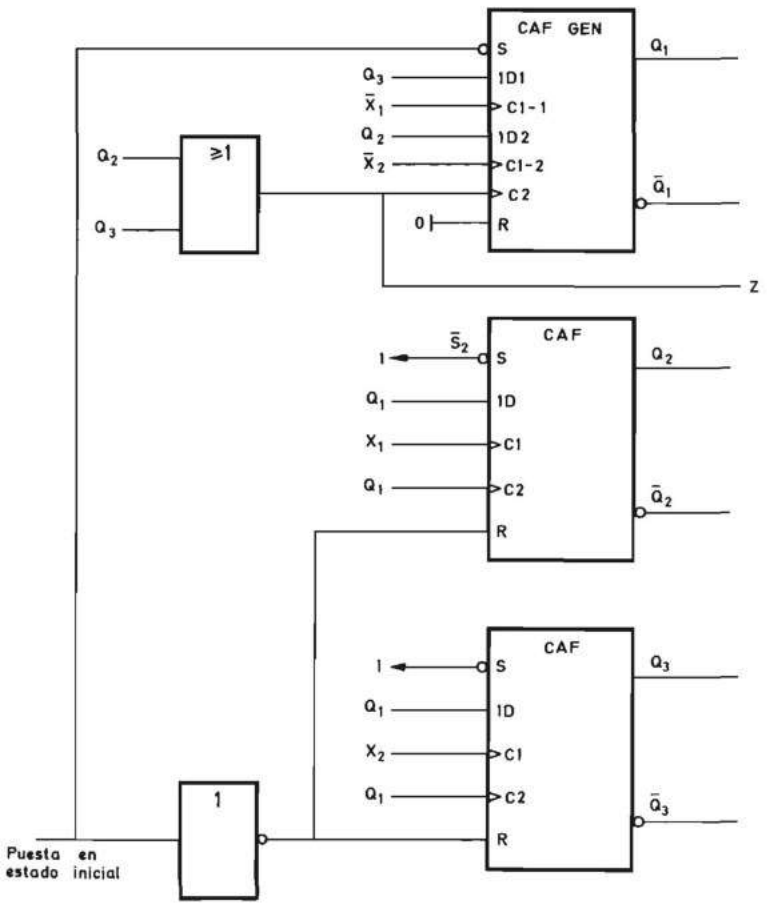


FIGURA 6.68.—**Esquema del circuito con células CAF correspondiente al autómata asíncrono del ejemplo 6.2.**

El esquema global se representa en la figura 6.68. Se supone que cada célula CAF posee una entrada de puesta a uno $\overline{S}$ (activa con un cero lógico) y una entrada de puesta a cero R (activa con un uno lógico). Mediante un nivel cero lógico aplicado a la entrada de puesta en estado inicial en el instante de dar tensión al autómata, se pone Q_1 en estado uno y Q_2 y Q_3 en estado cero. Una vez que desaparece dicho nivel, el autómata puede evolucionar de acuerdo con el grafo de transición de la figura 6.67.

La salida Z del autómata es igual a uno en los estados E_2 y E_3 y por ello se deduce:

$$Z = Q_2 + Q_3$$

Ejemplo 6.3:

En la figura 6.69 se repite el grafo de transición de la figura 6.53. Se propone

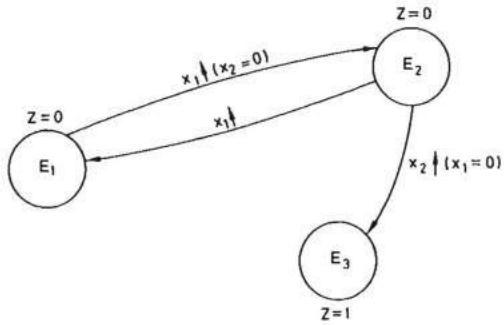


FIGURA 6.69.—**Grafo de transiciones del autómata asíncrono del ejemplo 6.3.**

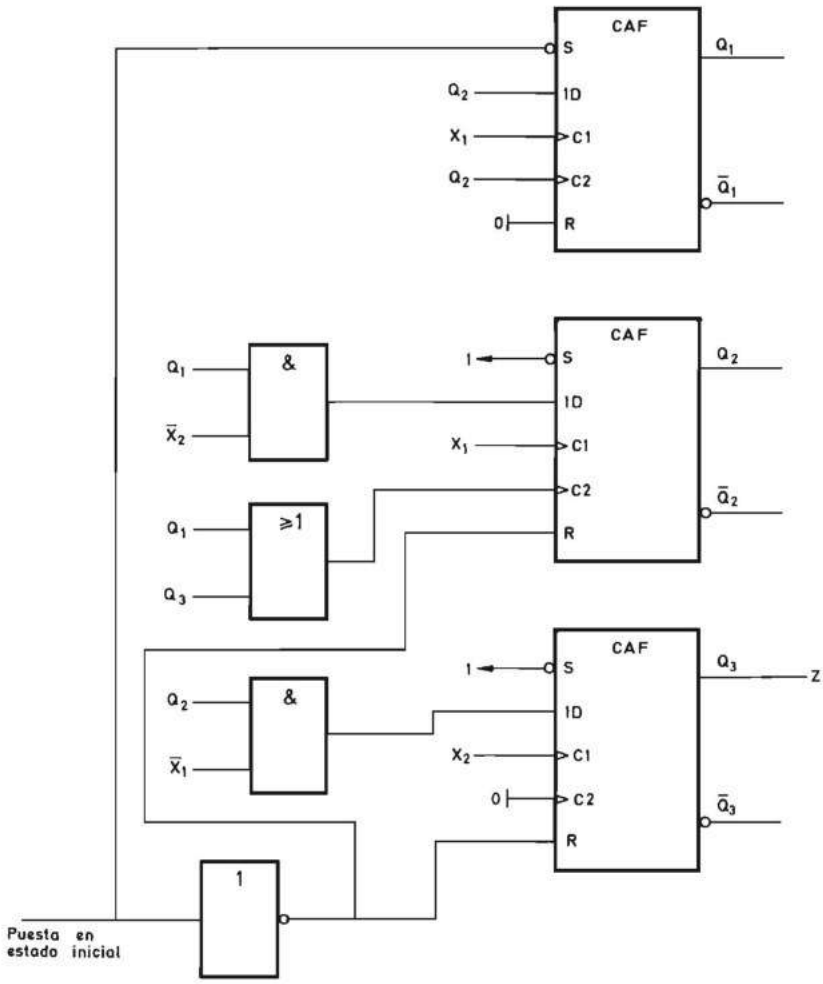


FIGURA 6.70.—**Esquema del circuito con células CAF correspondiente al autómata asíncrono del ejemplo 6.3.**

al lector que deduzca las ecuaciones de las entradas de cada una de las tres células CAF que constituyen el autómata:

Célula Q_1: $D_1\ (1D) = Q_2;\ T_1\ (C1) = x_1$
$T'_1\ (C2) = Q_2$

Célula Q_2: $D_2\ (1D) = Q_1\ \bar{x}_2;\ T_2\ (C1) = x_1$
$T'_2\ (C2) = Q_1 + Q_3$

Célula Q_3: $D_3\ (1D) = Q_2\ \bar{x}_1;\ T_3\ (C1) = x_2$
$T'_3\ (C2) = 0$

La entrada $T'_3\ (C2)$ de Q_3 se conecta al nivel cero porque desde E_3 el autómata no puede ir a ningún otro estado.

El esquema resultante se representa en la figura 6.70

6.2.3.5 Fenómenos aleatorios en los sistemas secuenciales asíncronos realizados con células activadas por flancos (CAF). Los fenómenos aleatorios en los sistemas secuenciales asíncronos diseñados por los métodos clásicos han sido ampliamente estudiados en la literatura sobre el tema y constituyen uno de los problemas que complican en mayor medida la síntesis.

Se analizan seguidamente los fenómenos aleatorios en la síntesis de autómatas asíncronos por el procedimiento desarrollado en apartados anteriores con células asíncronas activadas por flancos (CAF).

El fenómeno aleatorio se define como un comportamiento anómalo del autómata, que depende de las tolerancias en los parámetros físicos de los elementos utilizados en su realización. El comportamiento anómalo consiste en una transición entre estados no deseados. Las transiciones entre estados internos de transición se producen tal como se estudió en el apartado 6.2.3.1 al igualarse a uno la variable binaria capacidad de transición. El comportamiento correcto del autómata realizado con células activadas por flancos implica el que en cada instante una sola célula se encuentre en estado uno. Cuando uno de los productos de la capacidad de transición toma el valor uno, el autómata cambia de estado, para lo cual debe activarse una nueva célula y bajar al estado cero la que estaba activada. Un fenómeno aleatorio puede producirse si simultáneamente dos o más productos de la capacidad de transición adoptan el estado uno, lo cual producirá la activación de dos o más células. A continuación se analiza este fenómeno para las diferentes expresiones de la capacidad de transición. Supóngase un estado E_j de un autómata asíncrono de control y considérense los diferentes casos que pueden producirse:

a) A partir de E_j se pueden producir transiciones a los estados E_k y E_m si cambian de estado respectivamente las variables x_i o x_k (fig. 6.71*a*). Puede producirse el fenómeno aleatorio si ambas variables cambian de estado simultáneamente y se produce la subida a uno de Q_k y Q_m. Por lo tanto, este fenómeno no existe si ambas variables x_i y x_k no pueden cambiar simultáneamente.

b) Puede producirse otro fenómeno aleatorio si cambian simultáneamente una variable que actúa por el cambio de estado y otra que actúa por su nivel, lo cual

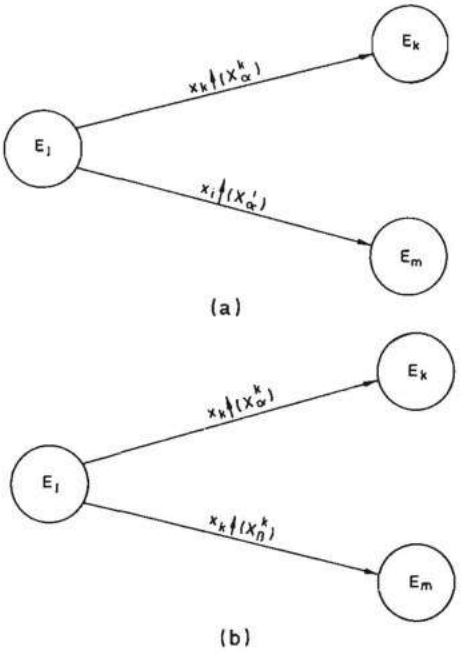


FIGURA 6.71.—**Ejemplos de grafos de transiciones con posibles fenómenos aleatorios.**

se representa en la figura 6.71*b*. El autómata debe pasar de E_j a E_k si cambia de estado $x_k(x_k \uparrow)$ y el vector formado por las demás variables de entrada coincide con X_α^k; por otra parte se debe producir la transición de E_j a E_m si cambia $x_k(x_k \uparrow)$ y el vector de entrada coincide con X_β^k. Si X_α^k difiere de X_β^k en el estado de una variable x_h y esta variable cambia de estado al mismo tiempo que x_k puede suceder que ambos términos de C_T, $x_k \uparrow (X_\alpha^k)$ y $x_k \uparrow (X_\beta^k)$ adopten el valor uno simultáneamente y se activan las dos células correspondientes a E_k y E_m. Esto puede evitarse, al igual que en el caso anterior, mediante el muestreo secuencial de x_k y x_h, lo cual implica elevar la complejidad del sistema.

Puede producirse otro fenómeno aleatorio debido a que los cambios de estado de las variables o vectores de entrada que actúan sobre las entradas T de las células CAF se realicen muy lentamente. Los fabricantes de los biestables activados por flancos utilizados en la realización de las células CAF especifican la máxima pendiente de las señales aplicadas a las entradas $C1$ y $C2$. Este problema se resuelve fácilmente conectando las variables de entrada x_i a través de circuitos disparador (trigger) de Schmitt, que por otra parte son necesarios en general, conjuntamente con elementos integradores, para anular los rebotes de aquéllas. En el libro «Manual de prácticas de electrónica digital» (cap. 6) del Departamento de Tecnología Electrónica de la Universidad de Vigo [MAND 95] se indican los circuitos correspondientes.

De esta forma sistemática quedan eliminados los fenómenos aleatorios en los autómatas asíncronos realizados con células CAF.

6.2.3.6 Problemática de la realización física de los sistemas secuenciales asíncronos. En apartados anteriores se analizó la síntesis de los sistemas secuenciales asíncronos caracterizados por niveles y por flancos. La caracterización por niveles presenta una problemática de aplicación que la hace inviable en la práctica. La caracterización por flancos ha permitido el desarrollo de métodos sistemáticos de diseño con células activadas por flancos analizados anteriormente para el caso de que sólo cambie una variable de entrada simultáneamente. Pero el método desarrollado adolece de algunos inconvenientes que se acentúan cuando se eleva el número de variables de entrada y salida:

a) El método de codificación de uno entre *n* eleva excesivamente la complejidad cuando el número de estados internos supera la decena.
b) Esa complejidad se eleva aún más en el caso, más usual en la práctica, de que varias variables de entrada cambien simultáneamente. Es necesario en este caso realizar una exploración secuencial de las variables de entrada.
c) Falta de modularidad, es decir, imposibilidad de que se pueda ampliar el número de variables de entrada o salida sin necesidad de cambiar el sistema físico.

Por todo lo expuesto el campo de aplicación de los sistemas secuenciales asíncronos es muy limitado.

Por otra parte, el progreso de la microelectrónica con el desarrollo de circuitos integrados a medida y semimedida está propiciando el desarrollo de controladores lógicos síncronos que superan todas las dificultades indicadas. A ellos se dedica el apartado 6.3.5.6.3 y al lector interesado en profundizar en su estudio se le remite a la bibliografía [INTE 88] [MAND 90].

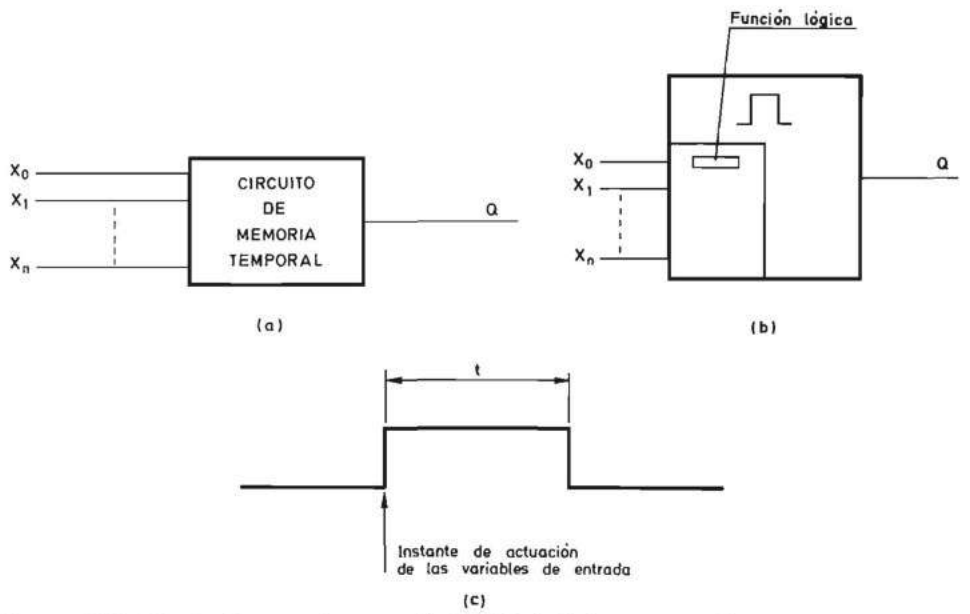


FIGURA 6.72—**Circuito de memoria temporal: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado; *c*) Evolución en tiempo de la salida *Q*.**

6.2.4. Circuitos digitales temporales

6.2.4.1. Introducción. En apartados anteriores se estudiaron los circuitos capaces de recordar las variaciones de una o más variables de entrada memorizándolas mediante el estado de una variable binaria. Dicha variable permanece en un determinado estado hasta que la variación de las mismas u otras variables de entrada la hacen volver al estado anterior. Los circuitos citados recibieron el nombre de biestables.

Pero en numerosas aplicaciones prácticas es conveniente que el retorno al estado inicial se realice al cabo de un cierto tiempo t variable según la situación particular. Los circuitos que actúan de esta manera reciben el nombre de circuitos de memoria temporal o temporizadores, porque memorizan una determinada situación mediante el estado de una variable binaria durante un cierto tiempo t y su diagrama de bloques se representa en la figura 6.72.

También resultan muy interesantes en la práctica los circuitos digitales que carecen de variables de entrada y cambian periódicamente de un estado al otro. Dichos circuitos reciben el nombre de osciladores de onda cuadrada o generadores de impulsos y en la figura 6.73 se representa su diagrama de bloques y la evolución en el tiempo de la señal que aparece en su salida Q.

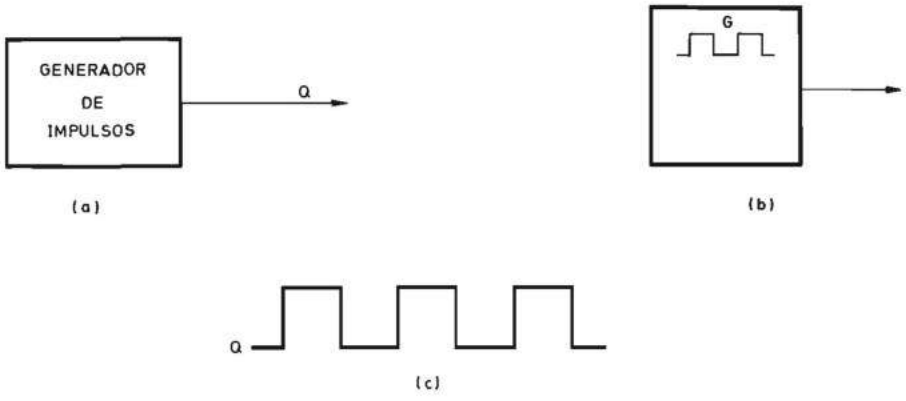


FIGURA 6.73.—**Generador de impulsos: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado; *c*) Evolución en el tiempo de la salida Q.**

6.2.4.2. Circuitos de memoria temporal. Se pueden a su vez clasificar en dos grandes clases: monoestables o temporizadores digitales y temporizadores analógico-digitales.

Los monoestables se realizan mediante puertas digitales, y los temporizadores analógico-digitales utilizan comparadores analógicos.

6.2.4.2.1 Monoestables. Estos circuitos presentan a su vez diversas variantes que se indican en la tabla 6.13.

Monoestables	No realimentados		no redisparables (non retriggerable)
	Realimentados	activados por niveles	redisparables (retriggerable)
		activados por flancos	

TABLA 6.13

Los monoestables no realimentados, como su nombre indica, son aquellos en los que la información se desplaza en un único sentido sin que exista una realimentación que dé lugar a una memorización.

Los monoestables realimentados son en realidad sistemas secuenciales con capacidad de memorización de las variaciones de las variables de entrada. Los monoestables realimentados se pueden considerar a su vez activados por flancos o por niveles, pero esta división no constituye un concepto útil porque el comportamiento de ambos es idéntico. Los monoestables activados por niveles generan un único impulso a la salida cada vez que su entrada alcanza un determinado nivel, independientemente del tiempo que permanezca en el mismo.

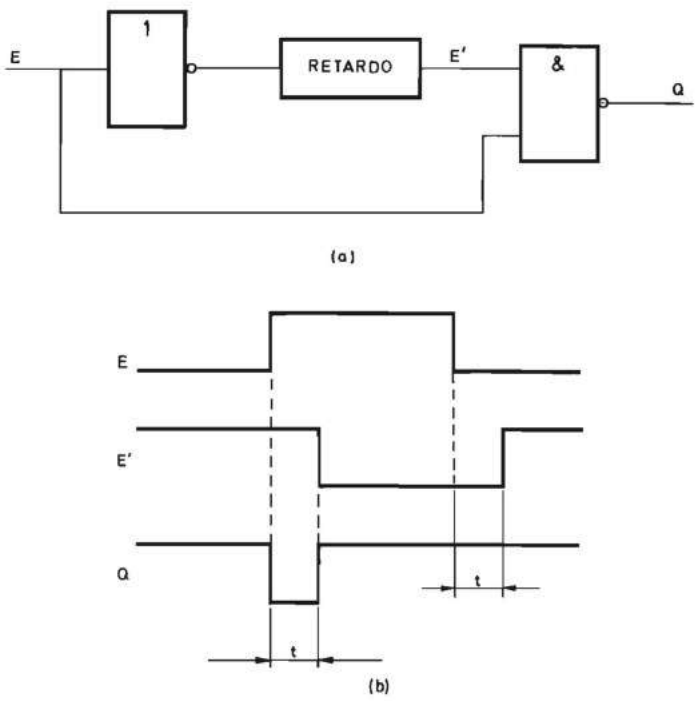


FIGURA 6.74.—**Monoestable no realimentado: *a*) Esquema típico; *b*) Evolución en el tiempo de sus variables.**

En sucesivos apartados se estudian los monoestables no realimentados y los realimentados activados por niveles.

6.2.4.2.1.1 Monoestables no realimentados. Estos monoestables se realizan mediante puertas interconectadas entre las que se introduce un retardo adecuado.

En la figura 6.74*a* se representa un esquema típico y en la figura 6.74*b* la evolución en el tiempo de sus principales variables. Si la variable E permanece indefinidamente en nivel cero, la salida Q se encuentra en nivel uno. Igualmente si E permanece indefinidamente en nivel uno, la variable E' se encuentra en nivel cero y, por tanto, Q también permanece en nivel uno.

Pero en el instante en que E pasa del nivel cero al uno la señal E' tarda un cierto tiempo t, igual al tiempo de propagación del inversor más el retardo, en pasar al nivel cero. Por ello las dos entradas de la puerta NO-Y (NAND) permanecen en nivel uno durante dicho intervalo t y, por tanto, en la salida Q aparece un impulso de esa duración, tal como se indica en la figura 6.74*b*.

El retardo de la figura 6.74*b* se puede obtener mediante una red RC tal como se indica en la figura 6.75.

En el caso de que la tecnología utilizada sea TTL, el valor de R no puede ser superior a 220 Ω para no disminuir demasiado la inmunidad al ruido de la puerta NO-Y (NAND). Otra manera de realizar físicamente el retardo es mediante un número par de inversores. En la figura 6.76 se representa con dos inversores.

Los monoestables activados por niveles sin realimentar se utilizan para generar impulsos de duración inferior al aplicado en su entrada y se aplican para simplificar las etapas de salida de los sistemas secuenciales síncronos cableados que se estudiarán más adelante.

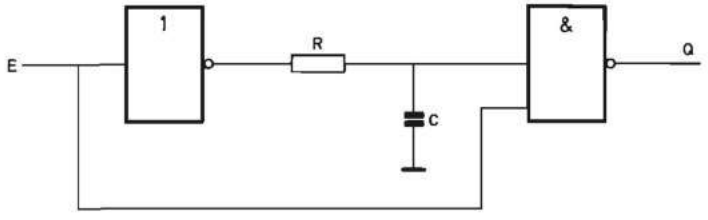


FIGURA 6.75.—**Esquema de un monoestable no realimentado con el retardo realizado mediante una red *RC*.**

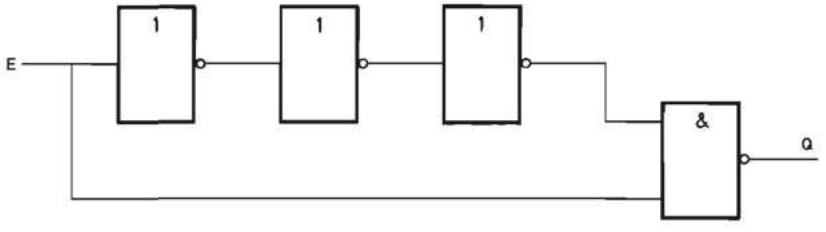


FIGURA 6.76.—**Esquema de un monoestable no realimentado con el retardo realizado con un número par de inversores.**

6.2.4.2.1.2 Monoestables realimentados. Los monoestables realimentados se pueden clasificar en dos grandes clases: activados por niveles, y activados por flancos (tabla 6.13).

Como su nombre indica, los monoestables activados por niveles son aquellos que generan un impulso de una cierta duración t cuando una o más variables de entrada adoptan un determinado nivel lógico que puede ser el cero o el uno lógicos.

Los monoestables activados por flancos generan el impulso cuando se produce un cambio de estado de una variable binaria. Dicho cambio puede ser del estado cero al uno lógico o viceversa. En el primer caso se dice que el monoestable es activado con flancos de subida y en el segundo caso, que lo es con flancos de bajada.

Como ya se indicó anteriormente, el comportamiento práctico de los monoestables activados por niveles y los activados por flancos es idéntico y por ello se estudiarán a continuación los primeros.

En la figura 6.77 se representa el biestable activado por niveles constituido por dos puertas NO-O (NOR) realimentadas estudiado en el apartado 6.2.2.2 (figura 6.22).

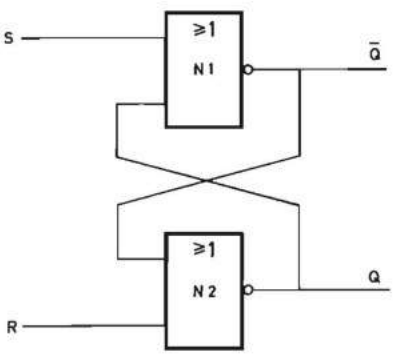


FIGURA 6.77.—**Biestable *R-S* activado por niveles realizado con dos puertas NO-O (NOR).**

Este biestable recuerda, mediante el estado de su variable de salida Q, cuál de las dos variables de entrada S o R ha sido la última en estar en nivel uno.

Modificando ligeramente el esquema de la figura 6.77 se obtiene el esquema del monoestable representado en la figura 6.78. Para ello se han unido entre sí las dos entradas de la puerta $N2$ y el punto común a través de la resistencia R al positivo de la tensión de alimentación. Dicho punto se une también a un condensador C cuyo otro extremo está conectado a la salida de $N1$.

Asignaremos a $+ V$ voltios el nivel uno lógico y a 0 voltios el nivel cero lógico. Supongamos que la entrada S se encuentra en el nivel cero al dar la tensión de alimentación a las puertas. Vamos a analizar lo que sucede tanto si la salida Q está a nivel cero como si está a nivel uno. Si la salida Q está a nivel cero la salida de $N1$ estará a nivel uno $(+ V)$, porque sus dos entradas están en nivel cero. El

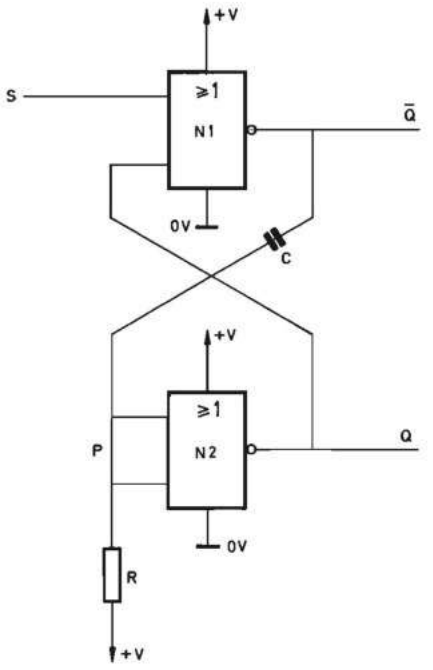


FIGURA 6.78.—**Monoestable activado por niveles realizado con dos puertas NO-O (NOR).**

condensador C estará descargado y el punto P a nivel uno, lo cual mantiene la salida Q a nivel cero. Se trata, por tanto, de un estado estable. Si por el contrario, Q está a nivel uno, la salida de $N1$ estará a nivel cero. Esto hará que el condensador inicie su carga. La tensión en el punto P es inicialmente cero y va elevándose hasta alcanzar el valor $+V$.

Mientras el nivel de P es inferior al del uno lógico, la salida Q se mantiene en nivel uno, pero en el instante en que lo alcanza, pasa al estado lógico cero. Por ello este estado es inestable. De lo expuesto se deduce que el circuito posee un solo estado estable y por ello recibe el nombre de monoestable.

En la figura 6.79 se representa el diagrama de evolución en el tiempo de las distintas variables del circuito de la figura 6.78 a partir del instante en que la variable S pasa del nivel cero al uno. La variable $\overline{Q}$ pasa al nivel cero y dicha transición se transmite al punto P a través del condensador C. Al bajar al nivel cero el punto P, la variable de salida Q pasa al nivel uno.

La carga del condensador produce la subida de forma exponencial de la tensión del punto P hasta alcanzar el nivel correspondiente al uno lógico (próximo a $+V$). Cuando la tensión en P alcanza el nivel del uno lógico, la salida Q vuelve al nivel cero. Esto hace que la tensión del punto P se incremente aproximadamente en $+V$, y que el condensador C se descargue a través de R.

La realimentación de la salida Q de la puerta $N2$ a una de las entradas de $N1$ hace que la duración del impulso obtenido en aquélla dependa sólo del valor de la red RC. Se pueden lograr así impulsos de duración superior al de entrada. Si

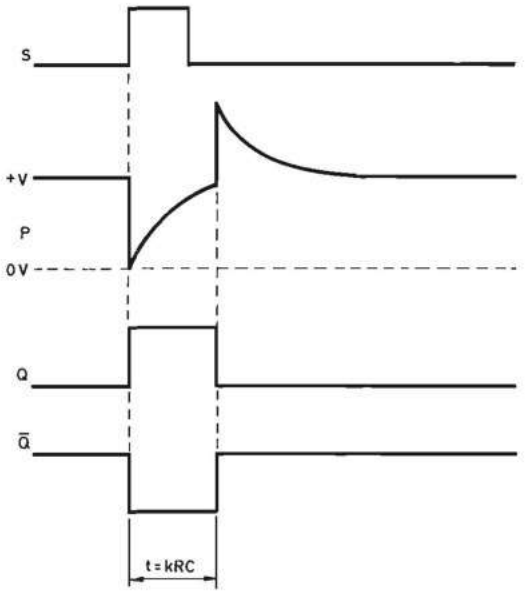


FIGURA 6.79.—**Diagrama de evolución temporal de las variables del monoestable de la figura 6.78.**

la variable de entrada S del circuito de la figura 6.78 se mantiene en nivel uno después de finalizar el impulso a la salida, Q se mantiene a nivel cero, el condensador C cargado, y por ello $\overline{Q}$ se mantiene también a nivel cero. Para que se produzca un nuevo impulso a la salida, es necesario que S vuelva al nivel cero y suba de nuevo al nivel uno.

El circuito monoestable que se acaba de estudiar constituye un bloque funcional que ha sido realizado en circuito integrado en diversas tecnologías con diferentes variantes. Por ello es interesante la representación de los monoestables mediante un diagrama de bloques indicado en la figura 6.80, en la que a la variable de entrada se le asigna la denominación E.

Tal como se estudió anteriormente, si la entrada E se encuentra a nivel cero

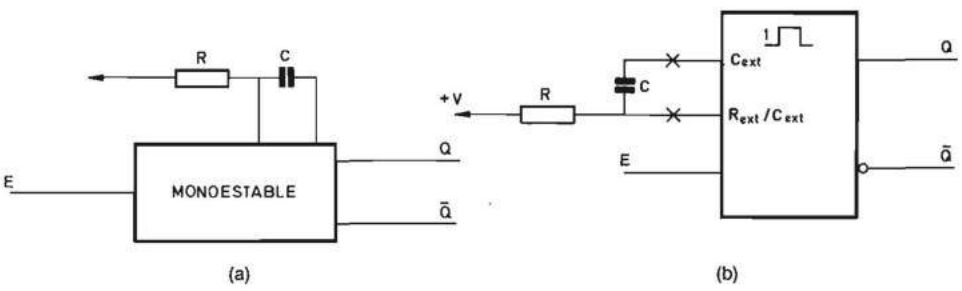


FIGURA 6.80.—**Diagramas de bloques de un monoestable no redisparable: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**

en el instante de dar tensión al circuito, el estado de la salida Q no está definido, lo cual es equivalente a que se puede producir un impulso en ella. Para evitarlo es posible dotar al monoestable de una entrada de puesta a cero (reset) que está unida a una entrada de la puerta $N2$. Si en el instante de dar tensión se aplica un nivel uno a la entrada de puesta a cero, se garantiza que no se produce un impulso en la salida Q.

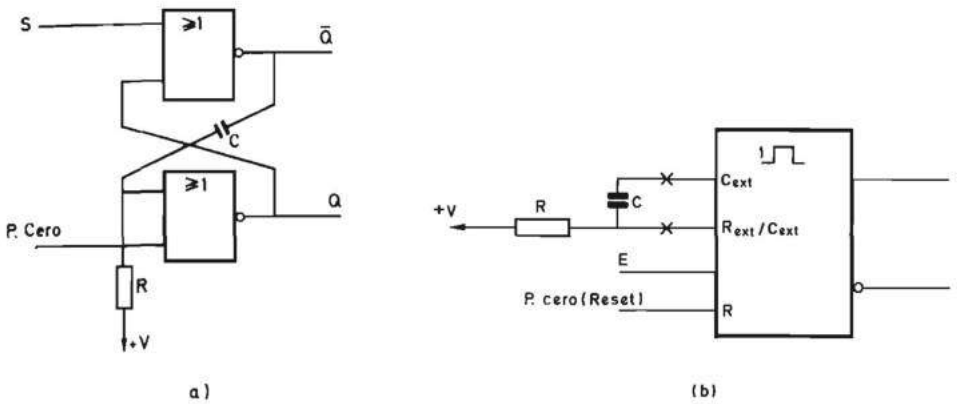


FIGURA 6.81.—**Monoestable no redisparable con entrada de puesta a cero: *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

En la figura 6.81 se representa el diagrama de bloques de un monoestable con entrada de puesta a cero activa con un nivel uno. La puesta a cero en el instante de dar tensión se puede realizar de forma automática mediante el sencillo circuito $R'C'$ y un inversor representados en la figura 6.82.

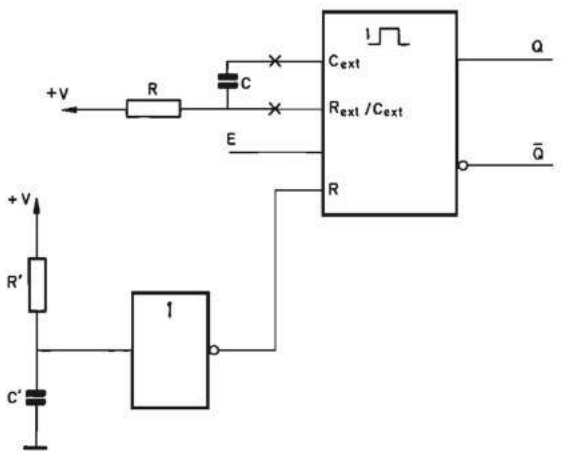


FIGURA 6.82.—**Monoestable con puesta a cero automática al dar tensión.**

El monoestable de la figura 6.78 se activa con el nivel uno de la variable de entrada *S*. Se obtiene un monoestable activo con el nivel cero sin más que añadir un inversor, tal como se muestra en la figura 6.83*a*. El diagrama de bloques se representa en la figura 6.83*b*, idéntica a la figura 6.81, a excepción de la entrada que tiene un círculo que indica que es activa con el nivel cero. En la figura 6.84 se representa el diagrama de secuencia de las señales $\overline{S}$ de entrada y Q de salida.

Los monoestables estudiados hasta ahora poseen una única variable de entrada. Pero, en ocasiones se desea generar un impulso si varias variables binarias pre-

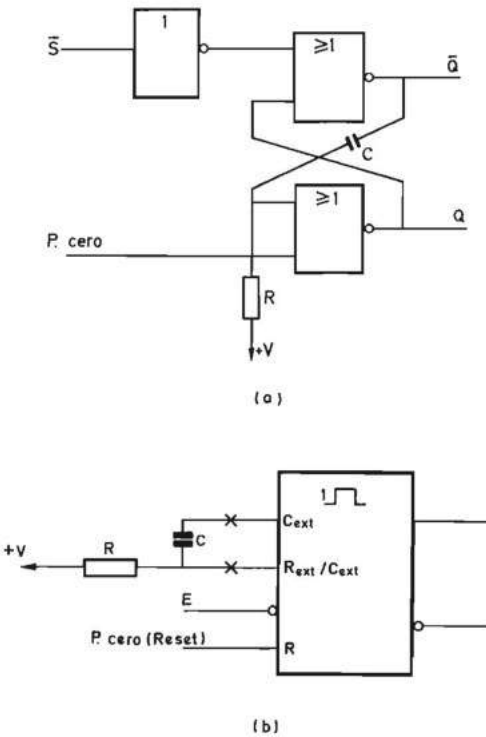


FIGURA 6.83.—**Monoestable activado mediante el nivel cero de la entrada: *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

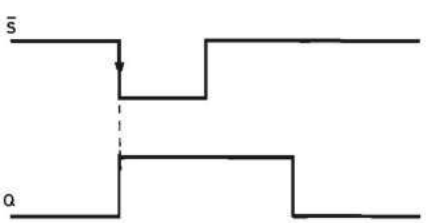


FIGURA 6.84.—**Diagrama de evolución temporal de las señales del monoestable de la figura 6.83.**

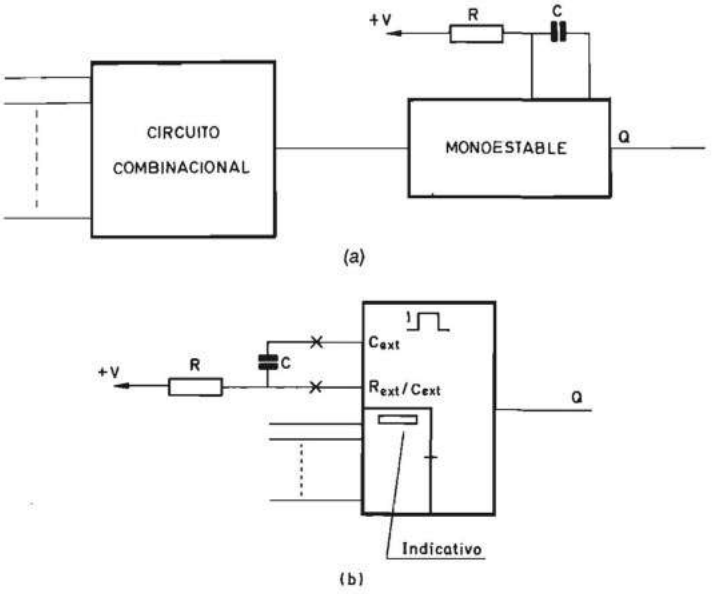


FIGURA 6.85.—**Monoestable con circuito combinacional a la entrada: *a*) Esquema con símbolos lógicos no normalizados; *b*) Símbolo lógico normalizado.**

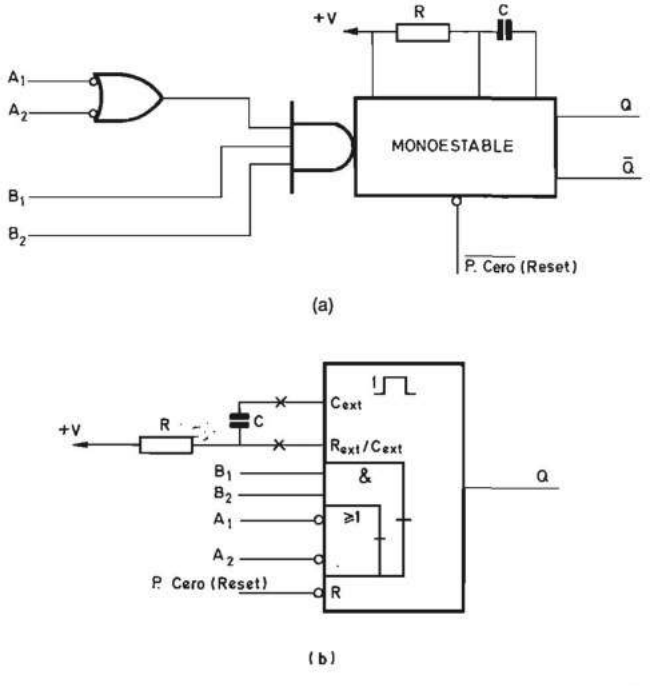


FIGURA 6.86.—**Ejemplo de monoestable con circuito combinacional a la entrada: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**

sentan una determinada combinación. Por ello, la práctica totalidad de los bloques funcionales integrados en las diversas tecnologías presentan un diagrama de bloques como el representado en la figura 6.85. En la figura 6.86 se representa un caso particular en el que el circuito combinacional realiza la siguiente función de las variables de entrada A_1, A_2, B_1 y B_2:

$$E = (\overline{A}_1 + \overline{A}_2)B_1B_2$$

El modo de operación del circuito se puede representar mediante una tabla de verdad como la de la tabla 6.14. La primera línea indica que la salida Q se mantiene a nivel cero si la entrada de puesta a cero ($\overline{\text{P. Cero}}$) se encuentra a nivel cero.

$\overline{\text{P. Cero}}$	A_1	A_2	B_1	B_2	Q	$\overline{Q}$
0	X	X	X	X	0	1
1	1	1	X	X	0	1
1	X	X	0	X	0	1
1	X	X	X	0	0	1
1	0	X	↓	1	⊓	⊔
1	0	X	1	↓	⊓	⊔
1	X	0	↓	1	⊓	⊔
1	X	0	1	↓	⊓	⊔
1	1	↑	1	1	⊓	⊔
1	↑	1	1	1	⊓	⊔
1	↑	↑	1	1	⊓	⊔

TABLA 6.14

Las tres líneas siguientes indican que, aunque la entrada de puesta a cero se encuentre en nivel uno, la salida Q se mantiene a nivel cero si la salida de la puerta Y se encuentra en ese estado, lo cual sucede si $A_1 = A_2 = 1$ (línea 2), o si $B_1 = 0$ (línea 3) o si $B_2 = 0$ (línea 4). Las restantes líneas indican las condiciones de disparo del biestable. La quinta línea, por ejemplo, indica que si no existe puesta a cero (P. Cero = 1), y además $B_2 = 1$ y $A_1 = 0$, se produce un impulso a la salida al cambiar B_1 de cero a uno.

Otra de las ventajas que presenta el circuito combinacional es la de permitir que el monoestable se pueda activar con niveles cero o uno, como el lector puede comprobar en la tabla 6.14.

Los circuitos monoestables que se acaban de estudiar presentan la característica de que si durante el intervalo t se aplica un nuevo impulso a la entrada E, el circuito lo ignora. En la figura 6.87 se representa de forma gráfica lo que se acaba de exponer. Los monoestables que se comportan de esta manera reciben la denominación de no redisparables (non retriggerable). Si el lector analiza el comportamiento del circuito de la figura 6.80, comprobará, en efecto, que una vez que se aplica

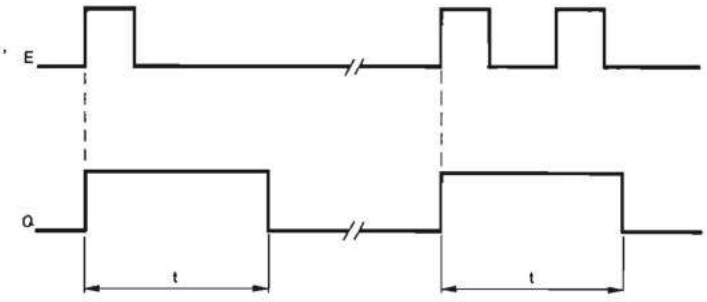


FIGURA 6.87.—**Diagrama de evolución temporal de las señales de un monoestable no redisparable (non retriggerable).**

un impulso a la entrada S de $N1$, baja al nivel cero la salida $\overline{Q}$ y se inicia la carga del condensador C. Mientras dura el impulso de salida en Q, la aplicación de un nuevo impulso en S no tiene efecto alguno.

Es posible la realización del monoestable de tal manera que cada impulso aplicado a la entrada descargue el condensador C y haga que se inicie de nuevo la temporización. El monoestable actúa de acuerdo con el diagrama de la figura 6.88, en el que se observa que la aplicación de un nuevo impulso a la entrada E durante la temporización hace que el impulso en la salida Q se prolongue un tiempo t. Estos monoestables reciben el nombre de redisparables (retriggerable).

En la figura 6.89 se representa un circuito monoestable redisparable. Cada vez que se aplica un impulso a la entrada S, el monoestable no realimentado, formado por las puertas $N3$ a $N6$, genera un impulso de corta duración que pone a nivel cero el punto P y produce la descarga del condensador C. Al mismo tiempo se pone a cero la salida $\overline{Q}$ y se inicia de nuevo la carga del condensador C, comenzando un nuevo intervalo de temporización. El diodo D es necesario si la puerta $N6$ posee una carga activa (pull-up) y tiene como misión evitar que C se cargue rápidamente cuando la salida de $N6$ pasa al nivel uno. En el caso de que $N6$ sea de colector abierto, D resulta superfluo.

Es conveniente resaltar que si el nivel de la entrada permanece activo una vez finalizado el intervalo de temporización, la salida del monoestable redisparable

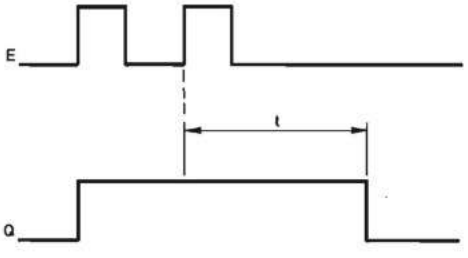


FIGURA 6.88.—**Diagrama de evolución temporal de las señales de un monoestable redisparable (retriggerable).**

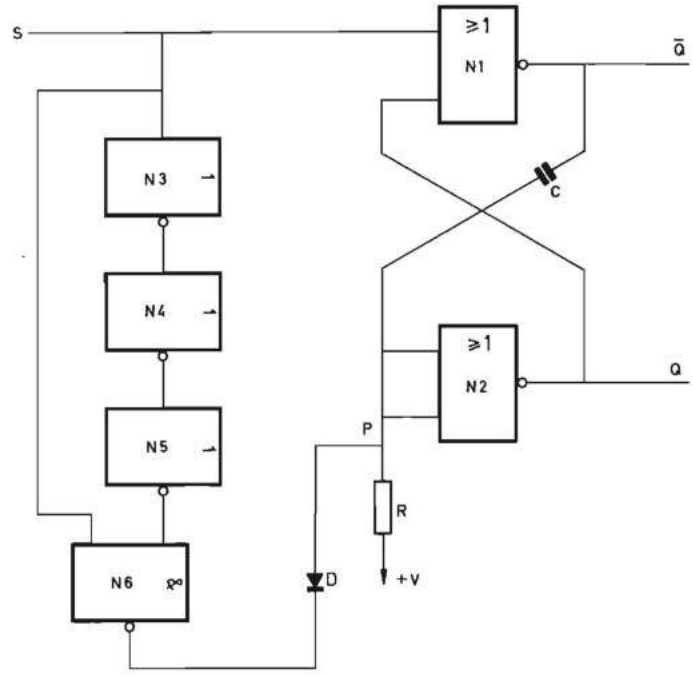


FIGURA 6.89.—**Esquema de un monoestable redisparable.**

baja a nivel cero y no vuelve al nivel uno hasta que la entrada se desactive y se vuelva a activar de nuevo. Este modo de operación es idéntico al de los monoestables no redisparables.

En la figura 6.90 se representa el diagrama de bloques de un monoestable redisparable. El símbolo no normalizado se distingue del correspondiente del monoestable no redisparable en que en su interior se coloca la denominación «monoestable redisparable». El símbolo normalizado se diferencia en que el número 1 no figura en la parte superior al lado del impulso.

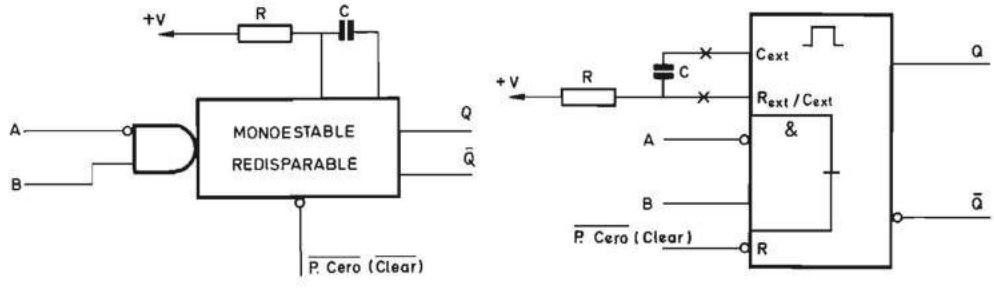


FIGURA 6.90.—**Ejemplo de diagrama de bloques de un monoestable redisparable con circuito combinacional de entrada: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**

6.2.4.2.2 Temporizadores analógico-digitales. Los monoestables estudiados en el apartado anterior constituidos por puertas lógicas realimentadas mediante un circuito resistencia-condensador no permiten lograr, en algunas tecnologías, temporizaciones superiores a algunos cientos de milisegundos, debido a la relativamente baja impedancia de las puertas lógicas (por ejemplo, en tecnología TTL). Combinando un comparador analógico con un biestable activado por flancos o por niveles se pueden realizar temporizaciones de varios minutos. Encadenando varios temporizadores o combinando un temporizador con un contador se pueden lograr temporizadores de varias horas.

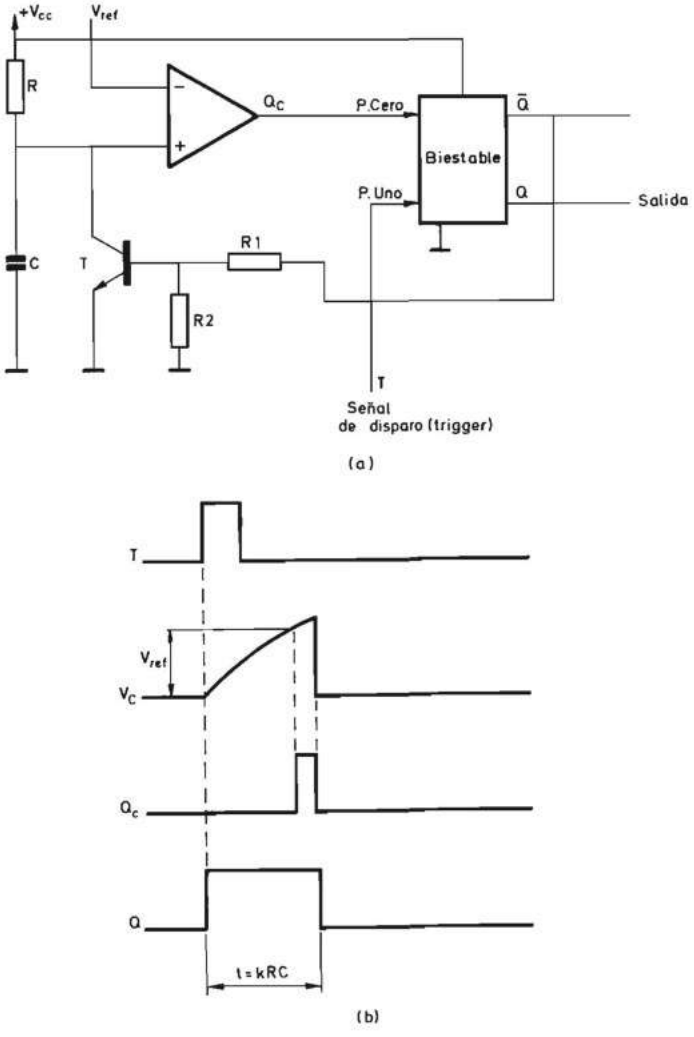


FIGURA 6.91.—**Esquema básico de un temporizador analógico-digital y diagrama de evolución temporal de sus señales.**

El comparador es un circuito analógico con dos entradas y una salida, que posee una elevada impedancia de entrada. La salida puede encontrarse solamente en dos estados diferenciados, según cuál de las entradas tenga un nivel de tensión superior a la otra. Para un estudio más detallado de los comparadores analógicos se remite el lector a la bibliografía [WAIT 85] [TAUB 80].

En la figura 6.91*a* se representa el esquema básico de un temporizador analógico formado, esencialmente, por un comparador analógico y un biestable activado por niveles (en este caso el nivel uno).

En la figura 6.91*b* se describe gráficamente la forma de operación de este circuito. Inicialmente Q se encuentra al nivel de cero voltios y $\overline{Q}$ al nivel de $+V$ voltios y el condensador C está cortocircuitado por el transistor T. Cuando se aplica un impulso a la entrada T del biestable, su salida Q pasa a nivel alto y la $\overline{Q}$ a bajo, con lo cual el transistor T se corta. Se inicia así la carga del condensador C a través de R_A hasta que V_C iguala la tensión de referencia. En este instante la salida Q_C del comparador pasa a nivel alto y pone de nuevo a nivel bajo la salida Q del biestable y a nivel alto la salida $\overline{Q}$.

La tensión de referencia V_{ref} se puede obtener a partir de $+V_{CC}$ mediante un divisor resistivo (fig. 6.92). El biestable utilizado puede ser de activación por niveles (apartado 6.2.2.2) o mediante cambios de nivel (apartado 6.2.3.3).

Para poder asegurar que no se produce un impulso a la salida del temporizador al darle la tensión de alimentación es conveniente dotar al biestable con una entrada de puesta a cero activa mediante un cierto nivel lógico. Esta entrada suele recibir el nombre de puesta en estado inicial. En la figura 6.92 se representa un temporizador analógico-digital con entrada de puesta en estado inicial activa con nivel cero ($\overline{\text{P. Inicial}}$).

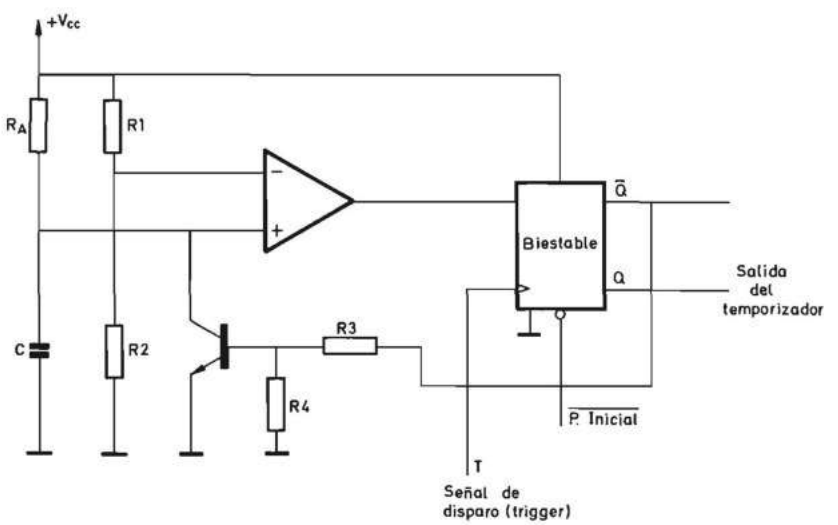


FIGURA 6.92.—**Diagrama de bloques de un temporizador analógico-digital activado por flancos de subida que posee una entrada de puesta en estado inicial.**

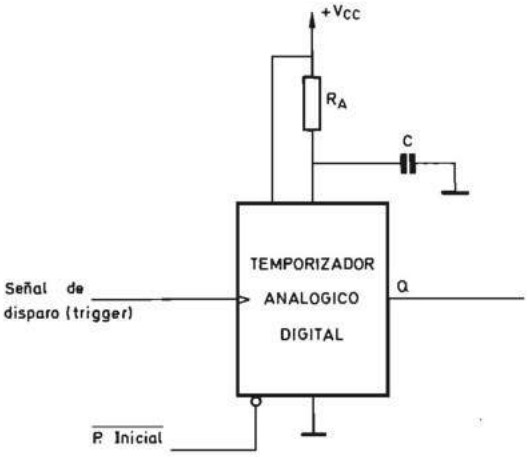


FIGURA 6.93.—**Diagrama de bloques del temporizador analógico-digital de la figura 6.92.**

Si el biestable utilizado es de activación por niveles, el impulso aplicado en la entrada de disparo T ha de ser de menor duración que el impulso generado en la salida Q porque, en caso contrario, ésta no bajaría a nivel cero al finalizar el intervalo de temporización. Para lograr un funcionamiento independiente de la duración del impulso de disparo se puede utilizar un biestable activado por flancos. El de la figura 6.92 se supone activo con flancos de subida (lo cual se indica mediante el símbolo $>$ colocado en la entrada correspondiente.

Si todos los componentes del circuito de la figura 6.92, a excepción de R_A y C, se colocan en un único circuito integrado, se puede representar aquél mediante el diagrama de bloques de la figura 6.93.

Utilizando como bloque básico el temporizador que se acaba de analizar es posible realizar circuitos complejos que constituyen bloques multifuncionales, que permiten realizar temporizadores y generadores de impulsos. Dichos circuitos se estudian en el apartado 6.2.4.3.2. En el apéndice 2 se describen diversos circuitos prácticos, desarrollados por varios fabricantes de acuerdo con la teoría expuesta en este apartado y en el 6.2.4.3.2.

6.2.4.3 Generadores de impulsos. Los generadores de impulsos u osciladores de onda cuadrada son circuitos electrónicos que carecen, en general, de señales de entrada y generan a su salida una onda cuadrada de una determinada frecuencia.

Los generadores de impulsos se pueden realizar mediante puertas adecuadamente realimentadas, constituyendo circuitos que reciben el nombre de astables. También es posible realizar generadores de impulsos mediante temporizadores analógico-digitales. En sucesivos apartados se estudiarán los diferentes circuitos.

6.2.4.3.1 Circuitos astables. Son generadores de impulsos formados por puertas lógicas adecuadamente realimentadas. Constituyen sistemas secuenciales que no poseen ningún estado estable.

En la figura 6.94 se representa el esquema más sencillo de un oscilador astable constituido por tres inversores conectados en serie de tal manera que la salida del tercero está unida a la entrada del primero. La conexión en serie de un número impar de inversores constituye un generador de impulsos.

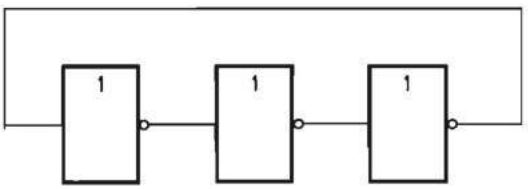


FIGURA 6.94.—**Generador de impulsos realizado con tres inversores.**

La duración del intervalo durante el cual la salida del generador de impulsos de la figura 6.94 permanece en nivel cero o uno es igual a tres veces el tiempo de propagación t_{pd} de cada puerta y, por tanto, la frecuencia de oscilación es $1/6\ t_{pd}$.

Pero tal como se estudió en el apartado 5.2, el tiempo de propagación de una puerta no es el mismo para los cambios de cero a uno que para los de uno a cero en su entrada y, por otra parte, ambos parámetros pueden variar de una puerta a otra dentro de unos márgenes definidos por el fabricante. De lo expuesto se deduce que el diseñador no puede decidir la frecuencia exacta de oscilación y que de acuerdo con el valor de los tiempos de propagación, aquélla varía aleatoriamente entre dos valores extremos.

El inconveniente al que se acaba de hacer referencia ha impulsado el desarrollo de circuitos con puertas realimentadas a través de redes resistencia-condensador o cristales de cuarzo, que permiten al diseñador seleccionar la frecuencia exacta de oscilación. Existen varias alternativas de las que se analizarán las más utilizadas.

En la figura 6.95 se representa el esquema de un generador de impulsos obtenido a partir del monoestable de la figura 6.88.

Para transformar el circuito de la figura 6.88 en el de la figura 6.95 se ha eliminado la entrada S y se ha unido la salida $\overline{Q}$ con la entrada de la puerta $N1$ a través de una red RC idéntica a la que enlaza la salida Q y la entrada de $N2$.

Supóngase que inicialmente Q se encuentra a nivel alto y $\overline{Q}$ a nivel bajo. En ese momento se inicia la carga de C_1 y la tensión en el punto B comienza a elevarse. En el instante en que su valor rebasa el nivel correspondiente al uno lógico (lógica positiva), la salida de $N1$ conmuta al nivel cero lógico y el punto A pasa a dicho nivel, lo cual conmuta $\overline{Q}$ al nivel uno. En ese instante se inicia la carga de C_2 y cuando el punto A alcanza el nivel lógico uno, $\overline{Q}$ vuelve a nivel cero y Q al uno. El ciclo descrito se repite indefinidamente. Los diodos D_1 y D_2 en paralelo con R_1 y R_2 impiden que la tensión en A y B alcance un valor igual a dos veces la tensión $+V$. En la figura 6.96 se representa el diagrama de secuencia en el tiempo de las señales en los terminales Q y $\overline{Q}$ y los puntos A y B.

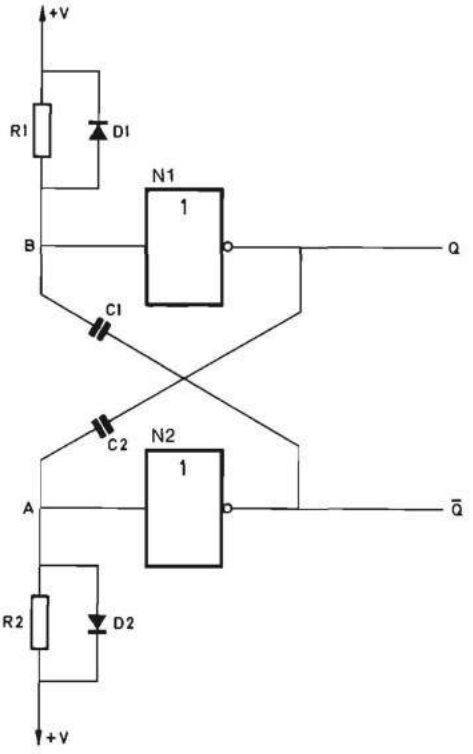


FIGURA 6.95.—**Generador de impulsos realizado mediante dos inversores realimentados a través de una red *RC*.**

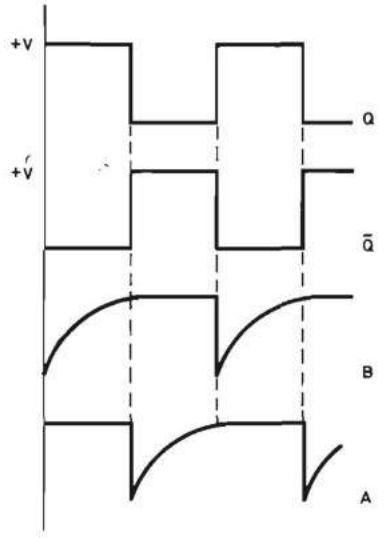


FIGURA 6.96.—**Diagrama de evolución temporal de las señales del generador de impulsos de la figura 6.95.**

El lector entenderá fácilmente que en el instante de dar tensión es posible que las dos salidas Q y $\overline{Q}$ se queden ambas en nivel cero, debido a que C_1 y C_2 se carguen a la misma velocidad. Una de las formas de evitarlo es que las resistencias R_1 y R_2 posean valores diferentes, lo cual presenta el inconveniente de que la señal obtenida a la salida del generador no constituye una onda cuadrada porque no permanece en nivel cero y uno el mismo intervalo de tiempo.

Otra forma de realizar un generador de impulsos es mediante la conexión de dos monoestables de tal manera que la finalización del impulso de uno de ellos provoque el inicio del impulso del otro y viceversa.

En la figura 6.97*a* se representa el esquema del circuito obtenido con dos monoestables activos con el nivel uno que pueden ser redisparables o no redisparables. La salida $\overline{Q}_1$ del primer monoestable se conecta a la entrada E_2 del segundo y la salida $\overline{Q}_2$ de éste a la entrada E_1 del primero. Supóngase que inicialmente el primer monoestable no está generando un impulso y que, por tanto, su salida Q_1 se encuentra a nivel cero, y su salida $\overline{Q}_1$ en nivel uno. En ese instante E_2 se encuentra en nivel uno y se produce la subida de Q_2 a nivel uno y la bajada de $\overline{Q}_2$ a nivel cero durante un intervalo t_2 fijado por R_2 y C_2. Transcurrido el tiempo t_2, Q_2 vuelve a nivel cero y $\overline{Q}_2$ al nivel uno y, por tanto, E_1 conmuta al nivel uno y se inicia la generación de un impulso en la salida Q_1, cuya duración t_1 depende de R_1 y C_1.

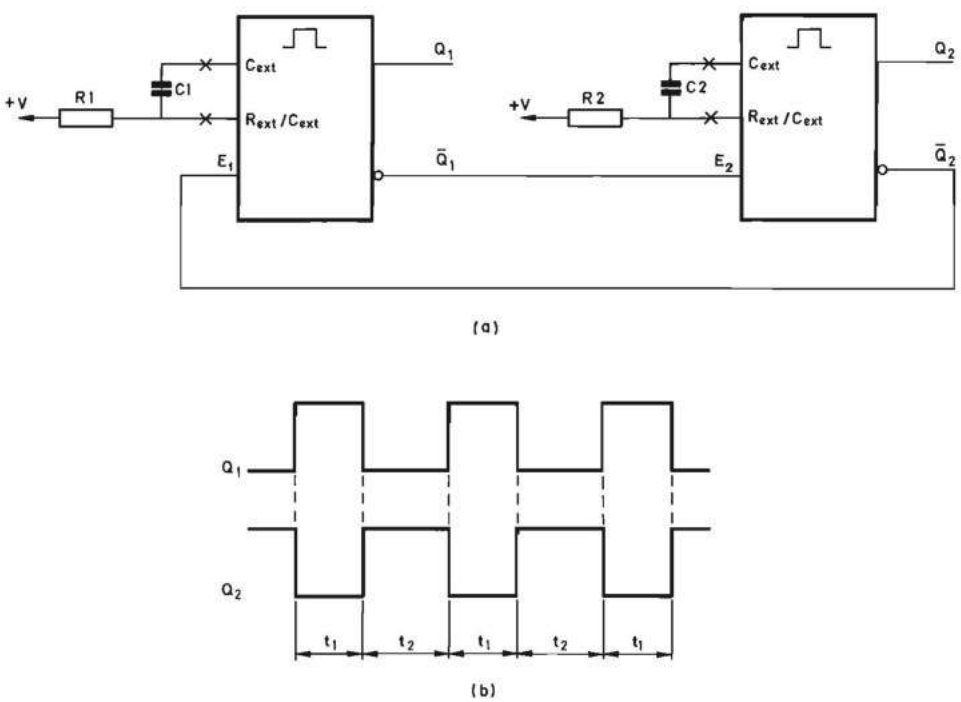


FIGURA 6.97.—**Generador de impulsos realizado mediante dos monoestables conectados en serie: *a*) Esquema realizado con símbolos normalizados; *b*) Diagrama de evolución temporal de sus señales.**

En la figura 6.97*b* se representa el diagrama de secuencia de la señal obtenida a las salidas Q_1 y Q_2 de ambos monoestables, cualquiera de las cuales puede ser considerada la salida del generador de impulsos. Haciendo R_1 igual a R_2 y C_1 igual a C_2 se tiene t_1 igual a t_2 y el circuito genera una onda cuadrada.

Si se utiliza un monoestable que posea un circuito combinacional a la entrada existen diversas posibilidades de conexión. Analícese como ejemplo un monoestable como el representado en la figura 6.98.

En la figura 6.99 se representa el esquema obtenido conectando Q_1 a $\overline{A}_2$ y Q_2 a $\overline{A}_1$. En la figura 6.100 se muestra otra alternativa obtenida uniendo $\overline{Q}_1$ a B_2 y $\overline{Q}_2$ a B_1.

En la figura 6.101*a* se representa el esquema de otro generador de impulsos realizado con monoestables que pueden ser no redisparables o redisparables. El primer monoestable constituye un oscilador y genera una señal cuya permanencia en nivel uno depende de R_1 y C_1, mientras que en nivel cero permanece un intervalo de tiempo muy pequeño igual al que tarda el monoestable en volver a dispararse (fig. 6.101*b*). Para obtener una onda cuadrada se conecta un segundo monoestable al primero con unos valores de R_2 y C_2 tales que su producto sea aproximadamente la mitad de R_1 C_1. Se obtiene así en Q_2 una señal que permanece el mismo

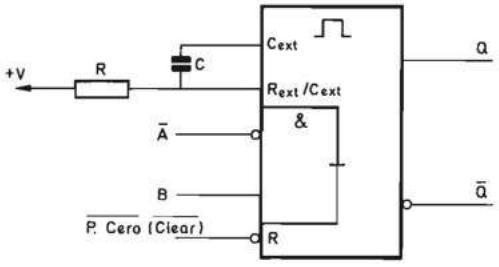


FIGURA 6.98.—**Símbolo lógico normalizado de un monoestable con circuito combinacional de entrada.**

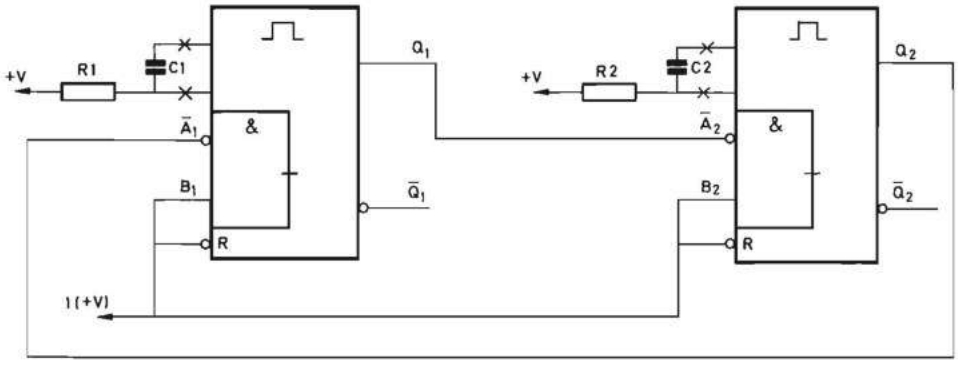


FIGURA 6.99.—**Variante de generador de impulsos realizado mediante la conexión en serie de dos monoestables idénticos al de la figura 6.98.**

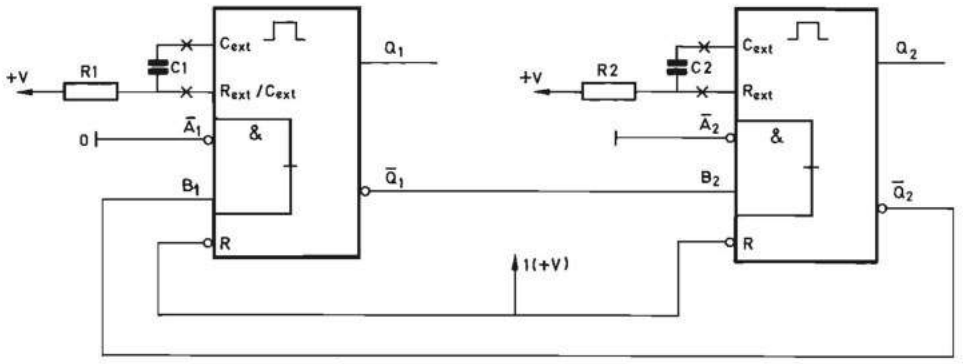
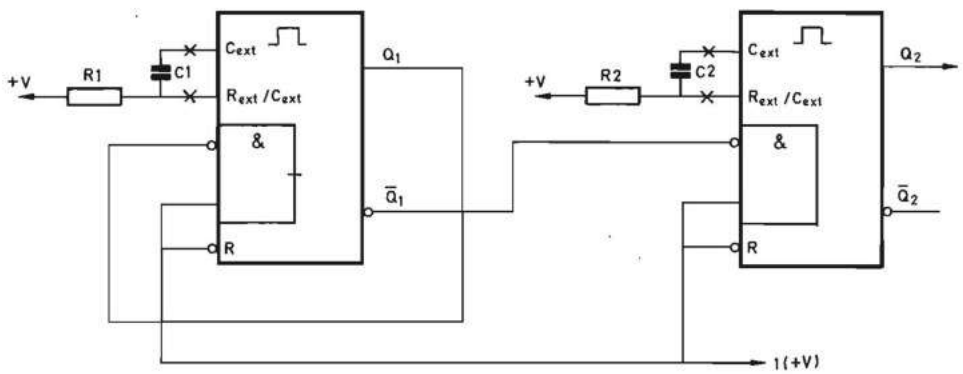
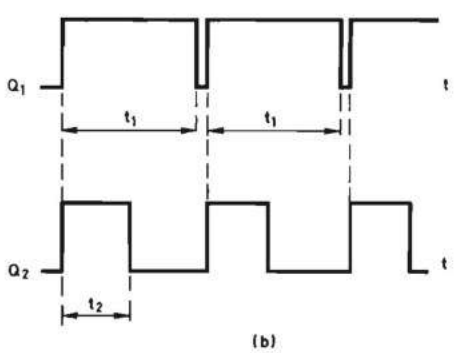


FIGURA 6.100.—**Variante de generador de impulsos realizado mediante la conexión en serie de dos monoestables idénticos al de la figura 6.98.**



(a)



(b)

FIGURA 6.101.—**Generador de impulsos realizado con dos monoestables: *a*) Esquema del circuito con símbolos lógicos normalizados; *b*) Diagrama de evolución temporal de sus señales.**

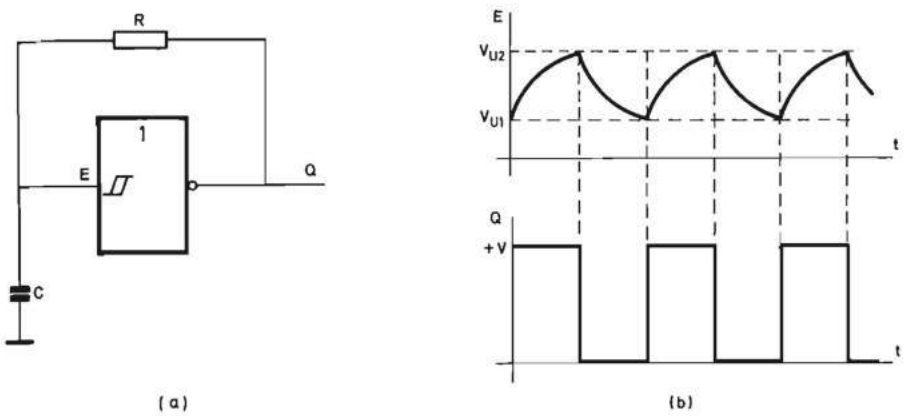


FIGURA 6.102.—**Generador de impulsos realizado mediante un inversor con histéresis (Schmitt trigger): *a*) Esquema del circuito; *b*) Diagrama de evolución temporal de sus señales.**

tiempo en nivel cero que en uno y que constituye, por tanto, una onda cuadrada (fig. 6.101*b*).

En la figura 6.102*a* se representa el esquema de un oscilador realizado mediante una puerta lógica inversora con histéresis (circuito disparador Schmitt). Las puertas lógicas inversoras normales estudiadas en el capítulo 5 (apartado 5.2) poseen una curva de transferencia como la de la figura 5.1 de tal manera que tanto cuando V_e se eleva como cuando disminuye, la tensión de salida V_s pasa por los mismos valores para cada uno de los valores de V_e. Mediante la adecuada configuración circuital [MILL 84] es posible realizar una puerta lógica con una curva de transferencia como la representada en la figura 6.103.

En la figura 6.102*b* se representa el diagrama de la evolución en el tiempo de las señales a la entrada y a la salida del circuito inversor con histéresis. Supongamos que inicialmente el condensador C está descargado y que la tensión en sus bornes es igual o inferior a V_{u1}; de acuerdo con la curva de transferencia de la figura 6.103, la salida Q del inversor con histéresis será aproximadamente igual a

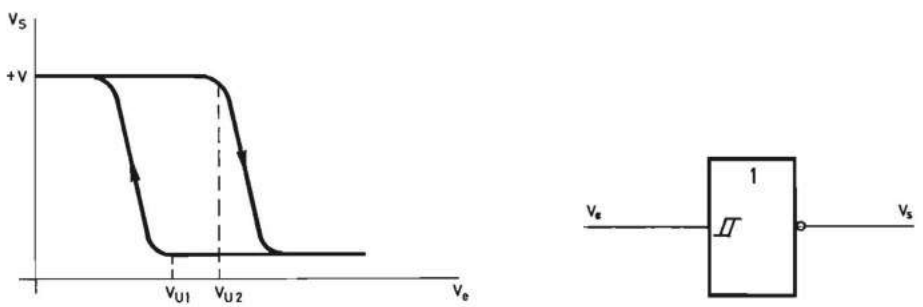


FIGURA 6.103.—**Curva de transferencia de un inversor con histéresis.**

$+V$, y se inicia la carga del condensador C a través de la resistencia R. En el instante en que la tensión en bornes del condensador alcanza el valor V_{u2}, se produce la conmutación de la salida Q al nivel aproximadamente igual a cero.

En ese instante se inicia la descarga de C a través de R hasta que la tensión en sus bornes alcance el nivel V_{u1}, instante en el que la salida Q vuelve al nivel uno, comenzando de nuevo el ciclo.

La figura 6.104 representa el esquema general de un circuito oscilador constituido por un amplificador realimentado a través de un circuito pasivo. Si el amplificador es lineal, a su salida se tendrá una onda senoidal cuya frecuencia será aquella a la que el desfase total sea nulo. Para que se produzca la oscilación es necesario además que la ganancia global del bucle a esa frecuencia sea igual a la unidad.

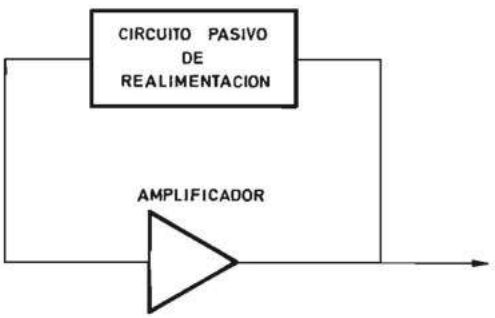


FIGURA 6.104.—**Esquema general de un oscilador realizado con un amplificador realimentado.**

Si el amplificador es no lineal y está constituido por una o más puertas lógicas, el oscilador generará una onda cuadrada. El circuito de realimentación puede estar constituido por una red RC o incluir además un cristal de cuarzo. En este último caso se logra un generador de onda cuadrada de gran estabilidad en su frecuencia de oscilación.

En la figura 6.105 se representa el esquema de un generador de onda cuadrada realizado mediante dos puertas inversoras y una red RC, típico en tecnología CMOS [NATI 84*a*].

Supóngase que inicialmente la salida Q se encuentra a nivel cero y el condensador C descargado. El punto A está a nivel cero y, por tanto, el punto B a nivel uno ($+V_{cc}$). Se inicia en ese instante la carga del condensador C y la tensión en el punto A comienza a crecer exponencialmente; en el instante en que la tensión en el punto A alcanza el nivel del umbral del uno lógico, la salida B del primer inversor conmuta al nivel cero y la salida Q del segundo inversor a uno ($+V_{cc}$). Esto hace que la tensión del punto A se eleve hasta $+V_{cc}$ y que se inicie seguidamente la descarga de C a través de R. En el momento en que el nivel de tensión disminuya por debajo del nivel de cero lógico, B vuelve al nivel uno, Q al cero y se inicia un nuevo ciclo de oscilación. La limitación de la tensión en el

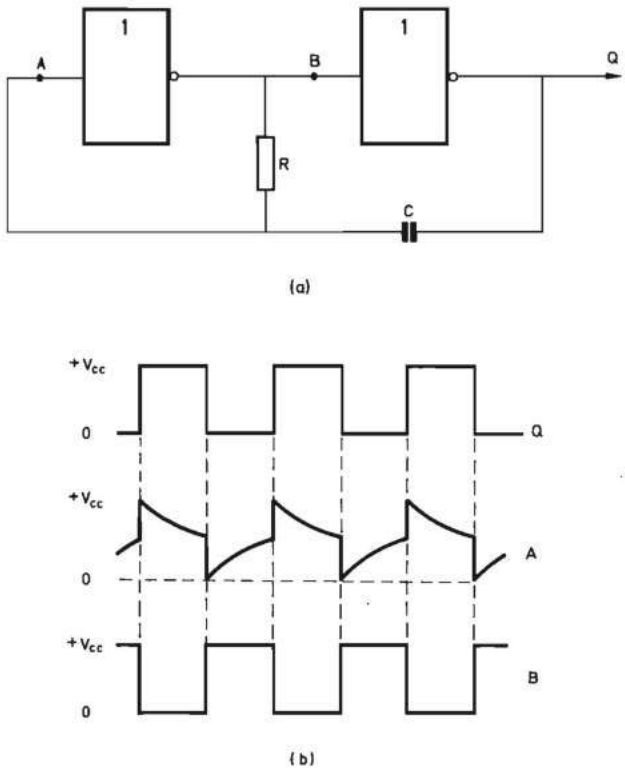


FIGURA 6.105.—**Generador de impulsos realizado con dos inversores realimentados mediante una red *RC*: *a*) Esquema del circuito; *b*) Diagrama de evolución temporal de sus señales.**

punto A entre $+V_{cc}$ y 0 se logra gracias a los diodos de protección conectados a la entrada de las puertas de tecnología CMOS [MAND 88*b*].

En la figura 6.106 se representa un generador de onda cuadrada con realimentación a través de un circuito RC y un cristal de cuarzo, típico en tecnología

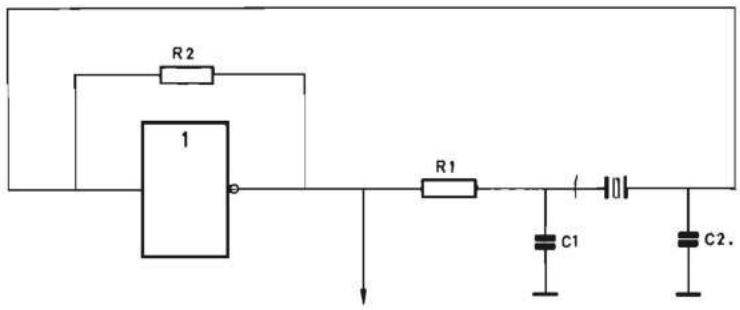


FIGURA 6.106.—**Generador de impulsos realizado con un inversor realimentado mediante una red *RC* y un cristal de cuarzo (tecnología CMOS).**

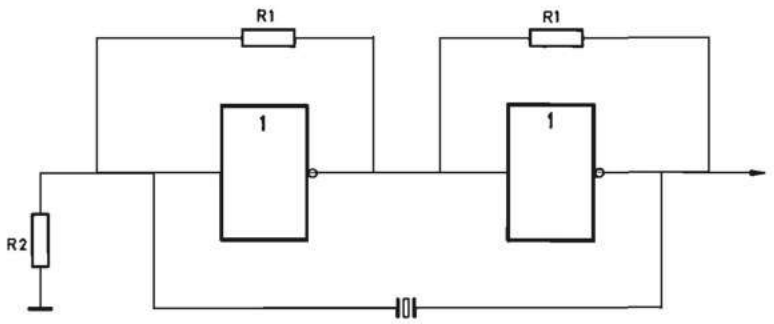


FIGURA 6.107.—**Generador de impulsos realizado con dos inversores realimentados mediante resistencias y un cristal de cuarzo (tecnología TTL).**

CMOS. En la figura 6.107 se representa otro circuito generador de impulsos utilizado en tecnología TTL similar al de la figura 6.105. Para un estudio más detallado de estos circuitos se remite al lector a la bibliografía [NATI 84*b*] [TOCC 77].

6.2.4.3.2 Generadores de impulsos con temporizadores analógico-digitales. En el apartado 6.2.4.2.2 se estudiaron los temporizadores analógico-digitales. La conexión de dos temporizadores analógico-digitales permite realizar un oscilador en onda cuadrada. En la figura 6.108*a* se representa el esquema correspondiente formado por dos temporizadores analógico-digitales activados por flancos de bajada. La salida del primero se conecta a la entrada de disparo del segundo y la salida de éste a la entrada de disparo del primero. En la figura 6.108*b* se representan las formas de onda obtenidas.

Por otra parte, complicando ligeramente el esquema de las figuras 6.91 y 6.92 se obtiene un circuito multifuncional que puede actuar como temporizador o generador de impulsos ampliando así su campo de aplicaciones. En la figura 6.109*a* se representa el esquema básico que contiene dos comparadores en lugar de uno solo. Las tensiones de referencia de ambos comparadores se obtienen a partir de la tensión $+V_{cc}$ por medio del divisor formado por tres resistencias de idéntico valor R. El punto común de las dos resistencias superiores se une a la entrada inversora (—) del comparador A, cuya salida está conectada a la puesta a uno del biestable y el punto común a las dos resistencias inferiores se une a la entrada no inversora (+) del comparador B, cuya salida está conectada a la puesta a cero del biestable. La carga del condensador C se realiza a través de las resistencias R_A y R_B cuando el transistor T está al corte. La descarga de C se realiza a través de B_B cuando el transistor T está saturado. La salida Q_T del temporizador se obtiene a partir de la salida Q del biestable a través de un amplificador inversor. El biestable posee una puesta en estado inicial que cuando se encuentra en un nivel activo pone a uno la salida Q del biestable y produce la descarga del condensador C (en la figura 6.109 se supone que es activada con el nivel cero).

Se describe seguidamente el funcionamiento que se ilustra gráficamente en la figura 6.109*b*.

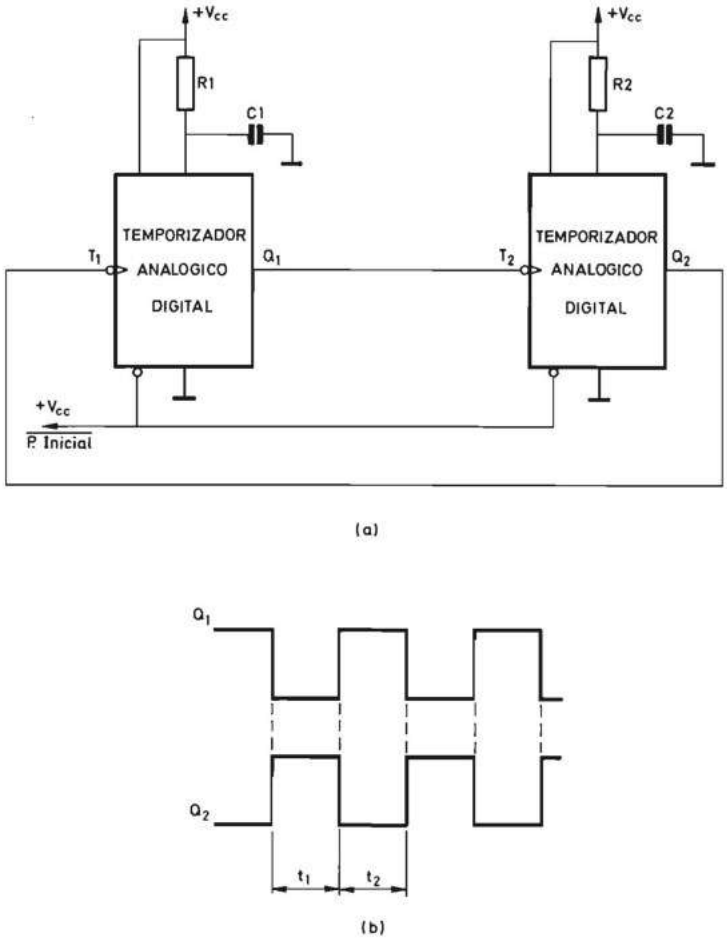
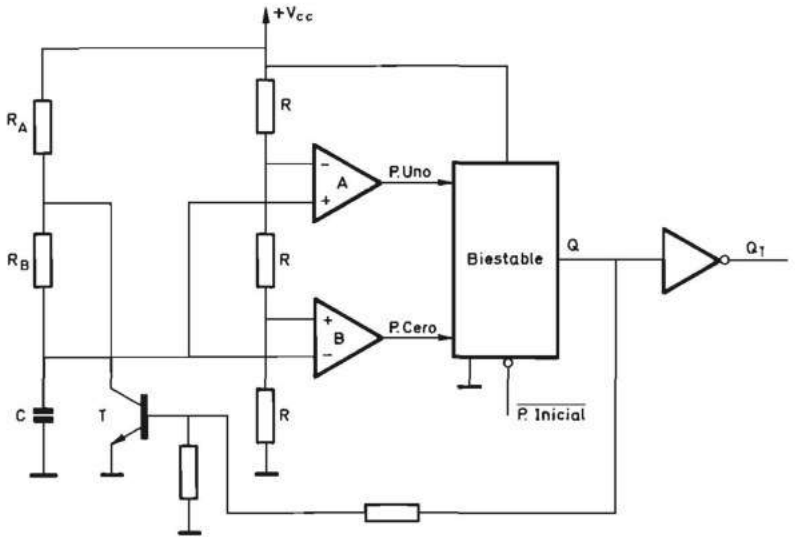


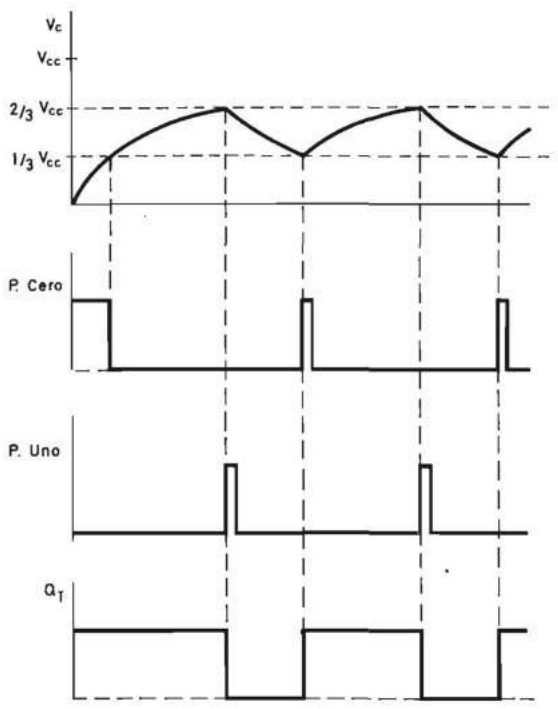
FIGURA 6.108.—**Generador de impulsos realizado mediante dos temporizadores analógico-digitales conectados en serie: *a*) Esquema del circuito; *b*) Diagrama de evolución temporal de sus señales.**

Supóngase que inicialmente se mantiene activada la puesta en estado inicial durante un cierto tiempo. La salida Q_T quedará en nivel cero (Q en nivel uno) y el condensador C descargado. Al desaparecer dicha puesta en estado inicial, la salida del comparador B, que se encuentra en nivel alto debido a que la entrada inversora (—) es menos positiva que la no inversora (+), hace que la salida Q del biestable pase a cero (Q_T se pone a nivel alto), y que se inicie la carga del condensador C. Cuando la tensión en bornes del condensador V_c supera el valor de la tercera parte de V_{cc} desaparece la puesta a cero del biestable porque la salida del comparador B cambia de estado.

Cuando V_c alcanza los dos tercios de V_{cc} pasa a nivel alto la salida del comparador A [su entrada no inversora (+) se hace más positiva que la inversora (—)] y se realiza así la puesta a uno del biestable con lo cual la salida Q_T del



(a)



(b)

FIGURA 6.109.—**Generador de impulsos realizado mediante un circuito temporal multifuncional analógico-digital: *a*) Esquema del circuito; *b*) Diagrama de evolución temporal de sus señales.**

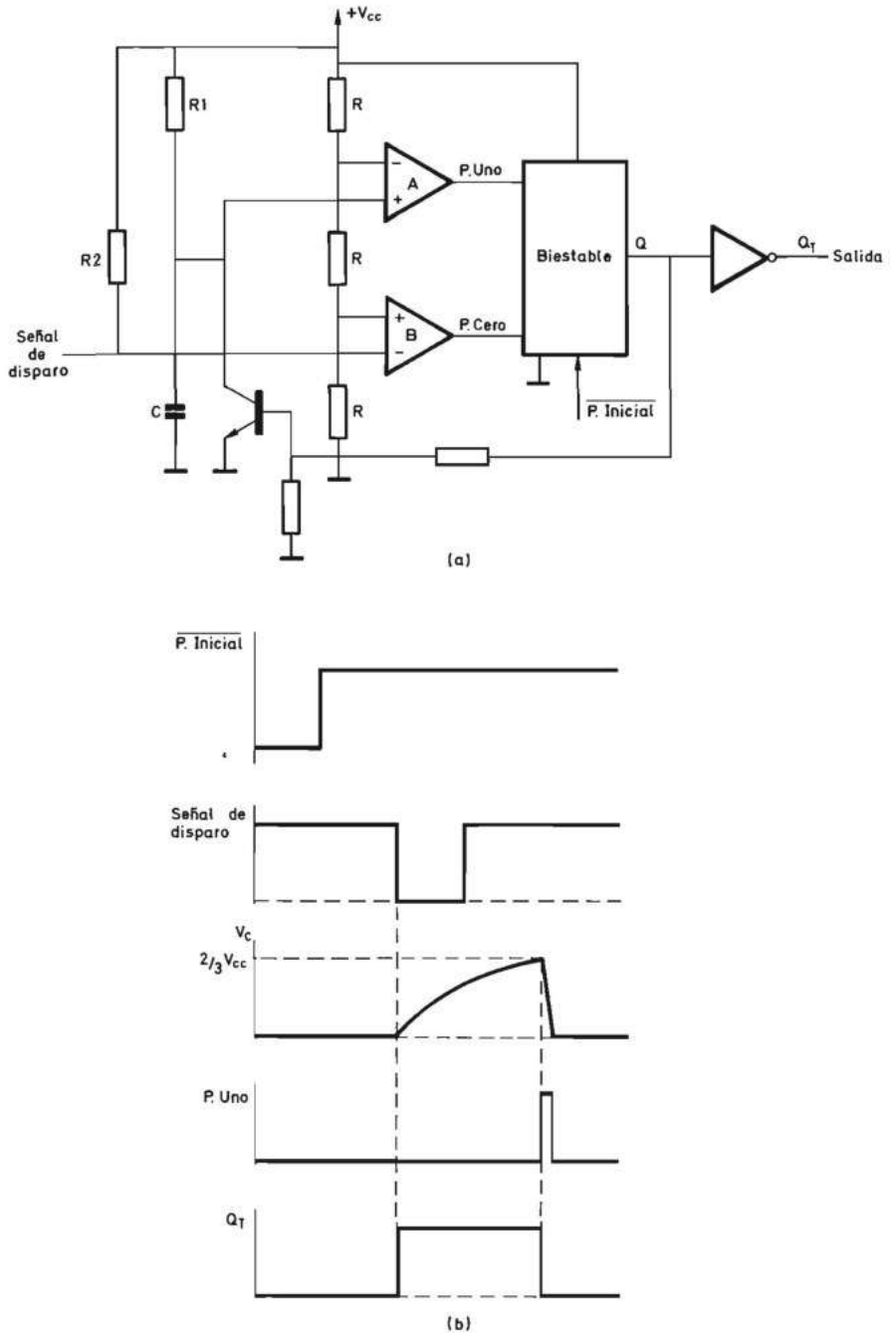


FIGURA 6.110.—**Temporizador realizado mediante un circuito temporal multifuncional analógico-digital: *a*) Esquema del circuito; *b*) Diagrama de evolucion temporal de sus señales.**

temporizador pasa a nivel cero y se inicia la descarga del condensador a través de R_B. Cuando V_c disminuye por debajo de $V_{cc}/3$, vuelve a pasar a nivel alto la salida del comparador B, se pone a cero la salida Q del biestable y se inicia un nuevo ciclo.

El circuito que se acaba de estudiar puede ser utilizado también como monoestable o temporizador. El esquema correspondiente se representa en la figura 6.110*a*. La entrada inversora (—) del comparador B se conecta a nivel alto ($+V_{cc}$) a través de una resistencia R_2, y constituye la entrada de la señal de disparo externa. La salida no inversora (+) del comparador A, se conecta al terminal común de la resistencia R_1 y el condensador C.

En la figura 6.110*b* se representa el cronograma de las señales del circuito. Se supone que en primer lugar se activa la puesta en estado inicial (en este caso activa con nivel bajo), que pone a uno la salida Q del biestable y provoca la descarga del condensador C. La bajada a nivel cero de la señal de disparo hace que pase a nivel alto la salida del comparador B, que genera una puesta a cero del biestable y, por tanto, el paso a nivel uno de Q_T. Se inicia de esta forma la carga del condensador hasta que su tensión alcanza los dos tercios de V_{cc}, instante en que pasa a nivel uno la salida del comparador A, que provoca la puesta a uno del biestable (Q_T pasa a cero) y la vuelta al estado normal. El sistema permanece en dicho estado mientras no se produce una nueva señal de disparo.

De lo expuesto se deduce que los elementos comunes a las figuras 6.109 y 6.110 constituyen un bloque funcional temporizador que puede ser convertido en un astable o en un monoestable conectándolo a los elementos externos adecuados.

En la figura 6.111 se representa el citado bloque funcional en el que para tener una adecuada versatilidad se conectan al exterior:

a) El colector del transistor T: Entrada de descarga (Discharge input).
b) El terminal inversor (—) del comparador B: Entrada de Disparo (Trigger-input).
c) El terminal no inversor (+) del comparador A: Entrada de umbral (Threshold input).
d) El terminal no inversor (—) del comparador A: Entrada de control (Control voltage input).

Este bloque funcional ha sido realizado en un único circuito integrado monolítico bajo la denominación de 555. En el apéndice 1 se representa su diagrama de bloques con indicación de los terminales y se analizan varios circuitos prácticos.

6.2.4.4 Aplicaciones de los circuitos digitales temporales. Los circuitos de memoria temporal tienen como principal aplicación la generación de impulsos de duración programable cuando una o más variables lógicas alcanzan unos niveles determinados. Para esta aplicación se pueden utilizar tanto monoestables redisparables como no redisparables.

Pero la combinación de monoestables con otros elementos lógicos como, por ejemplo, puertas, biestables, etc., da lugar a circuitos con aplicaciones muy diversas, de las que se analizan algunas a continuación.

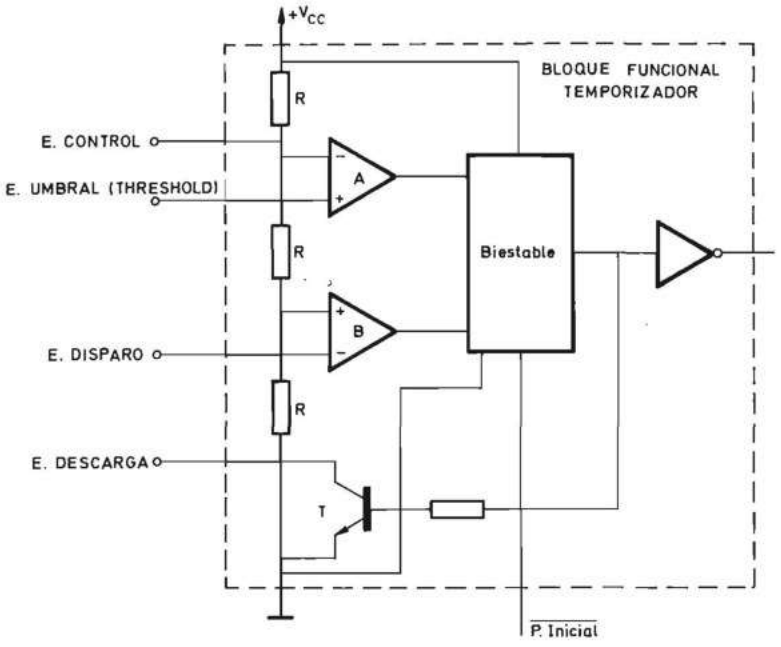


FIGURA 6.111.—**Bloque multifuncional temporizador analógico-digital.**

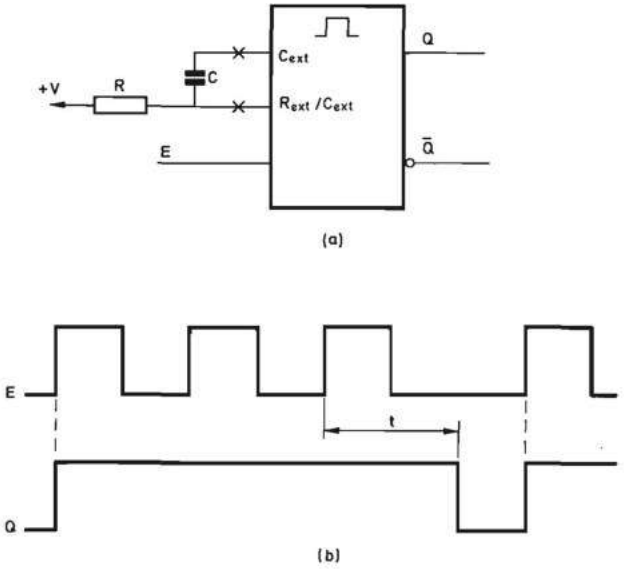


FIGURA 6.112.—**Detector de ausencia de impulsos: *a*) Esquema del circuito; *b*) Diagrama de evolución en el tiempo de sus señales.**

En la figura 6.112*a* se representa un circuito detector de ausencia de impulsos que constituye una de las principales aplicaciones de los monoestables redisparables. En la figura 6.112*b* se indica gráficamente la forma de funcionamiento.

Mientras la separación entre los impulsos aplicados a la entrada T sea inferior al tiempo t seleccionado mediante la red RC, la salida Q del monoestable se mantendrá en nivel uno.

Este circuito constituye también un detector de frecuencia mínima. En efecto, si a la entrada E del monoestable se aplica una secuencia de impulsos, la salida Q permanecerá en nivel uno si la frecuencia no disminuye por debajo de un cierto valor correspondiente a una separación entre impulsos que sea superior al tiempo t correspondiente a la red RC.

Otra aplicación típica de los monoestables es la conversión de la posición de un potenciómetro en la duración de un impulso, constituyendo un convertidor analógico-temporal. La duración del impulso se puede medir mediante un procesador digital que almacenará de esta forma, en una posición de su memoria, una combinación binaria cuyo valor es equivalente a la posición del potenciómetro. En la figura 6.113 se representa el esquema que consiste en un monoestable redisparable o no redisparable cuya resistencia R ha sido sustituida por un potenciómetro P.

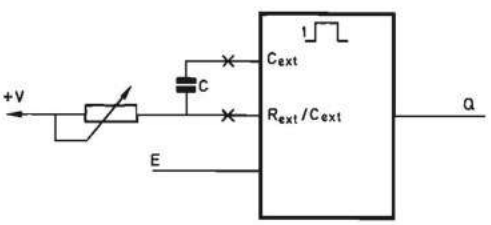


FIGURA 6.113.—**Circuito convertidor de la posición de un potenciómetro en la duración de un impulso.**

En la figura 6.114*a* se representa un circuito que detecta la anchura de los impulsos de entrada y sólo permite la aparición a su salida de un impulso cuando aquél rebasa un cierto valor. En la figura 6.114*b* se representa el diagrama de la evolución en el tiempo de las señales del circuito. La salida f es el producto lógico de la entrada E y la salida $\overline{Q}$ del monoestable y solamente alcanza el nivel uno si la duración del impulso aplicado a E es superior a la duración del impulso que aparece a la salida del monoestable programado mediante la red RC. Los dos inversores $N1$ y $N2$ tienen como misión evitar que el retraso en la conmutación del monoestable produzca un impulso de pequeña duración a la salida f cuando E pasa de cero a uno.

Los temporizadores analógico-digitales tienen aplicaciones idénticas a los monoestables, pero su característica de permitir la generación de impulsos de gran duración les hace especialmente útiles para la realización de temporizadores en aplicaciones industriales.

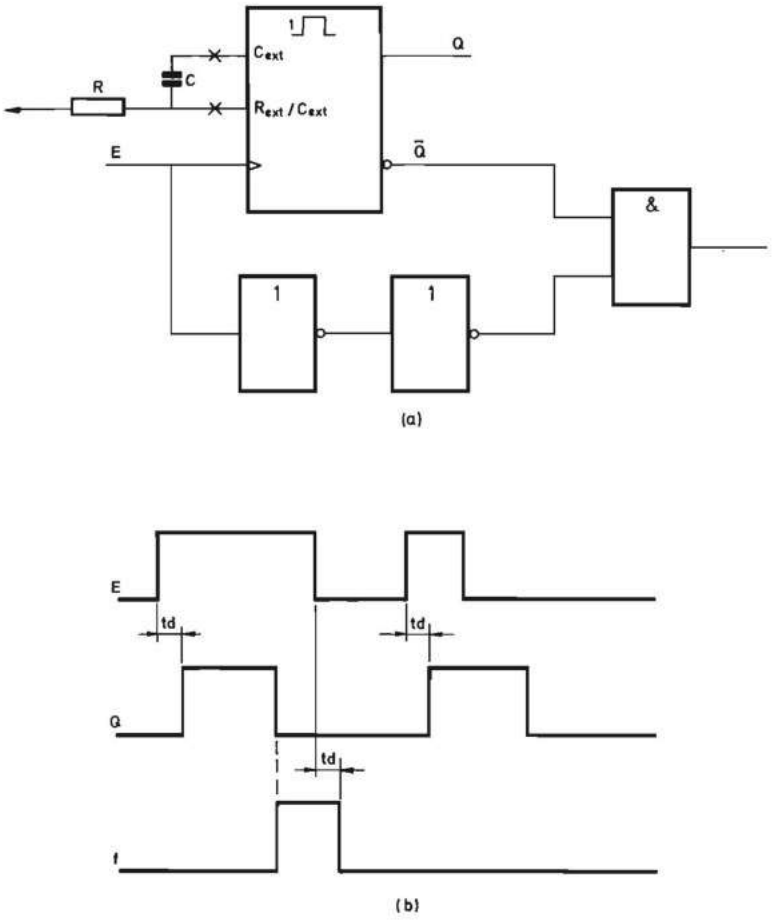


FIGURA 6.114.—**Circuito detector de anchura de impulsos: *a*) Esquema del circuito; *b*) Diagrama de evolución temporal de sus señales.**

En la figura 6.115 se representa una aplicación típica que combina un generador de impulsos realizado con un temporizador analógico-digital, un contador y un biestable. La señal de disparo pone a uno el biestable, eliminando de esta forma la puesta a cero del generador de impulsos y la inhibición del contador. A partir de ese instante el contador realiza el contaje de los impulsos del generador hasta alcanzar la máxima capacidad, lo cual viene indicado por la subida a uno de la salida Q del contador. Esta variable permanece en estado uno durante un ciclo del generador de impulsos, y su vuelta al estado cero hace que el biestable retorne a su estado inicial, quedando nuevamente puesto a cero el generador de impulsos e inhibido el contador hasta que se produzca una nueva señal de disparo.

El circuito de la figura 6.115 ha sido realizado en circuito integrado por diversos fabricantes [INTER 81] [MAXI 85].

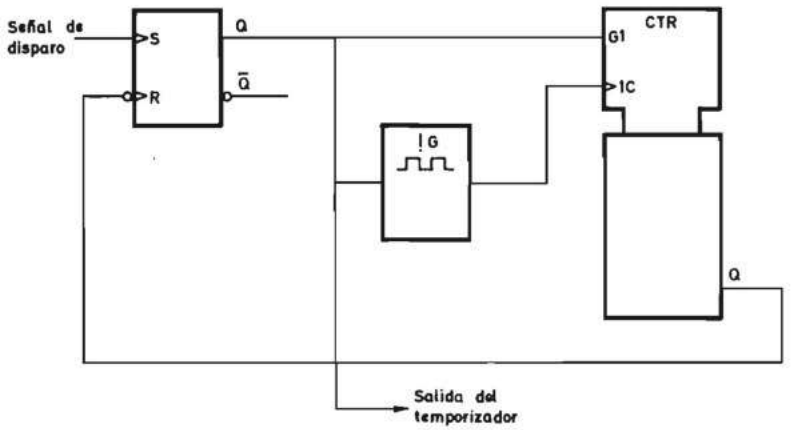


FIGURA 6.115.—**Temporizador realizado con un generador de impulsos y un contador.**

6.3 SISTEMAS SECUENCIALES SINCRONOS

6.3.1 Introducción

En apartados anteriores se han estudiado los sistemas secuenciales en los que las variables de entrada y las variables de estado interno actúan directamente sobre el sistema, dando lugar a un estado de salida y un nuevo estado interno. En estos sistemas es necesario evitar las transiciones no adyacentes de las variables de estado interno y las variables de salida.

Los sistemas secuenciales síncronos son aquellos en los que las variables de estado interno o las variables de entrada no actúan directamente sobre el sistema, sino que lo hacen en los instantes en que éste recibe impulsos procedentes de un generador.

Los impulsos del generador pueden actuar de dos maneras:

a) Por niveles, cuando permiten que las variables de entrada actúen sobre el sistema en el instante en que adoptan un determinado nivel activo que puede ser el cero o el uno lógicos.

b) Por flancos, cuando dicha acción se produce en el instante en que cambian de nivel. El cambio de nivel o flanco activo puede ser el de subida (cambio de cero a uno) o el de bajada (cambio de uno a cero).

En sucesivos apartados, se comprueba que los sistemas secuenciales síncronos realimentados, cuya estructura responde al esquema de la figura 6.6 que se repite en la figura 6.116, utilizan células básicas o biestables activados por flancos. Los sistemas secuenciales síncronos activados por niveles se utilizan exclusivamente para la memorización de información binaria.

En los sistemas secuenciales síncronos activados por flancos no existe el problema de las transiciones críticas siempre y cuando el tiempo que transcurra entre dos flancos activos sea superior al tiempo máximo de propagación del sistema combinacional (fig. 6.116).

Se iniciará el estudio de los sistemas secuenciales síncronos por el de los biestables síncronos o células binarias elementales síncronas. Se comprobará que su complejidad es superior a la de los biestables asíncronos activados por niveles, en particular el *R-S*. La aparición de los biestables síncronos en circuito integrado representó un gran avance para el diseño y realización de los sistemas secuenciales síncronos.

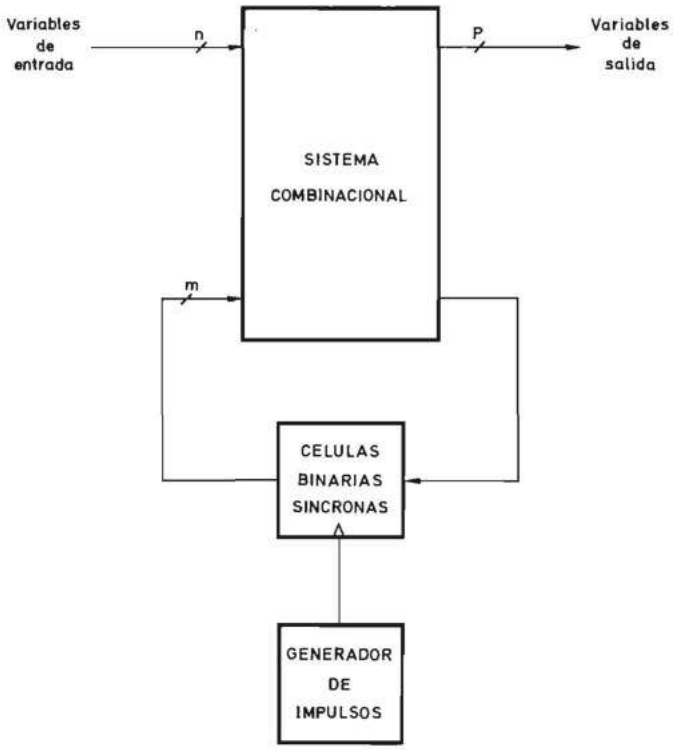


FIGURA 6.116.—**Diagrama de bloques de un sistema secuencial síncrono.**

A continuación se estudiarán los sistemas secuenciales síncronos de control, su constitución y forma de funcionamiento, y se observará que existen numerosos sistemas secuenciales síncronos que pueden ser utilizados como bloques básicos para la construcción de los demás.

Por ello reciben el nombre de bloques funcionales síncronos y han sido desarrollados en circuito integrado.

Seguidamente se elaborará una teoría general de los bloques funcionales síncronos, a cuya luz se analizarán los distintos casos particulares que han sido realizados en circuito integrado y se estudiarán los métodos de síntesis de sistemas secuenciales con ellos.

Finalmente se analizarán las aplicaciones de los sistemas secuenciales síncronos de control que se dividen en dos grandes clases:

a) Los sistemas secuenciales síncronos de control que reciben señales binarias externas y evolucionan entre estados internos en función del estado o cambios de estado de aquéllas. Estos sistemas se comportan igual que los sistemas secuenciales asíncronos con la ventaja de que superan los inconvenientes de éstos y reciben el nombre de controladores lógicos. A su estudio se dedica el apartado 6.3.5.6.3.
b) Los sistemas secuenciales síncronos de control que se utilizan conjuntamente con una unidad aritmética como las estudiadas en el capítulo 4. El sistema así obtenido recibe el nombre de procesador digital secuencial. La trascendencia de los procesadores digitales secuenciales que, gracias al progreso de las técnicas de integración, están revolucionando la electrónica aplicada, ha hecho que a ellos se les dedique el capítulo 8.

6.3.2 Biestables sincronizados

En el apartado 6.2.2.2 se han estudiado los biestables no sincronizados o asíncronos activados por niveles en los que las entradas controlan directamente su estado.

Los biestables sincronizados, cuyo diagrama de bloques se representa en la figura 6.117, poseen una entrada de impulsos de sincronización o de disparo (*T*) (trigger) y una o más entradas de información, cuyo estado lógico actúa sobre el biestable solamente cuando lo permite aquélla. La variable de salida del biestable recibe la denominación Q. A esta variable se la suele llamar salida directa. Normalmente se dispone también de su inversa que recibe el nombre de salida inversa y se designa $\overline{Q}$.

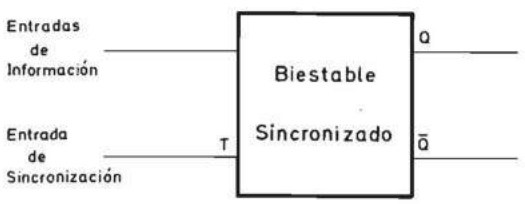


FIGURA 6.117.—**Biestable sincronizado.**

Según la forma en que la entrada de sincronización controla la actuación de las entradas de información, los biestables sincronizados son de dos tipos diferentes:

a) Biestables sincronizados por niveles. La entrada *T* permite que las entradas de información actúen sobre el estado del biestable durante todo el tiempo que se encuentra en un determinado nivel que se denomina activo.
b) Biestables sincronizados por cambios de nivel (flancos). El estado de las entradas de información actúa sobre el biestable en el instante en que se produce un flanco activo en la entrada *T*.

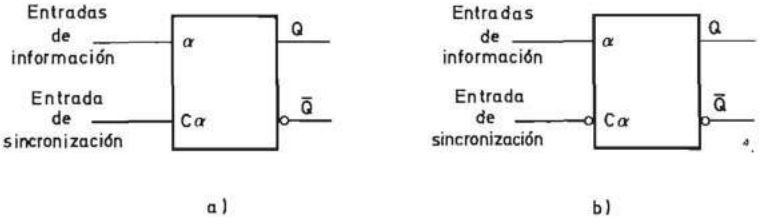


FIGURA 6.118.—**Símbolo lógico normalizado de los biestables sincronizados por niveles: *a*) Nivel uno; *b*) Nivel cero.**

La representación gráfica de los biestables sincronizados también se ha normalizado. La entrada de sincronización recibe el apelativo de C (control) seguido de un número decimal α y a las entradas de información se les asigna ese mismo número α.

Cuando la entrada C actúa por nivel no se le añade ningún otro indicativo, a excepción de un círculo si es activa con el nivel cero. En la figura 6.118 se representan los símbolos lógicos normalizados de sendos biestables sincronizados mediante niveles uno (fig. 6.118*a*) y cero (fig. 6.118*b*) lógicos.

En el caso de que la entrada C actúe por flancos se le añade un triángulo isósceles. En la figura 6.119 se representan los símbolos lógicos normalizados de los biestables sincronizados activados por flancos de subida (fig. 6.119*a*) y por flancos de bajada (fig. 6.119*b*).

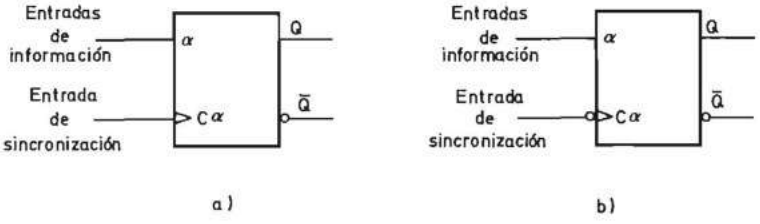


FIGURA 6.119.—**Símbolo lógico normalizado de los biestables sincronizados por flancos: *a*) Flanco de subida; *b*) Flanco de bajada.**

Existen varias formas de realizar los biestables sincronizados y, en particular, los activados por flancos. En la tabla 6.15 se representa una clasificación de los principales biestables sincronizados que se estudian en apartados sucesivos.

El diseño de estos biestables se puede realizar por los métodos indicados en el apartado 6.2.2.1 o bien mediante biestables no sincronizados activados por niveles, en particular el tipo *R-S*.

La complejidad de los biestables sincronizados, en especial de los activados por flancos, es relativamente elevada y su realización mediante componentes discretos implica la utilización de gran cantidad de éstos. Por esto, en la práctica, la utilización de los biestables sincronizados se desarrolló con la aparición de los circuitos integrados.

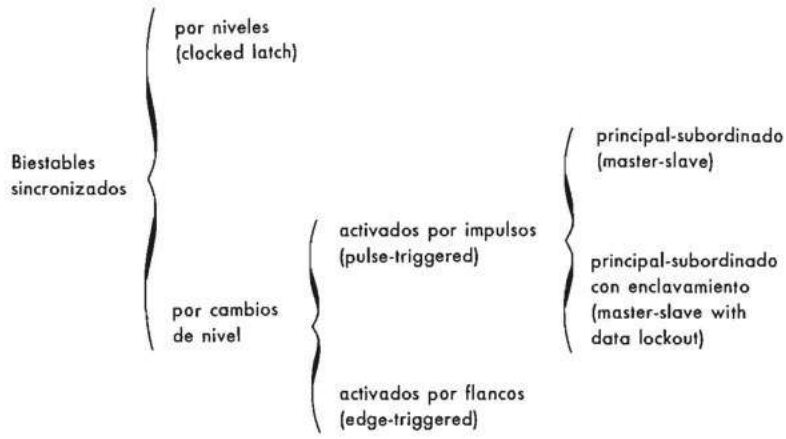


TABLA 6.15

6.3.2.1 Biestables sincronizados por niveles. Se realizan muy fácilmente con biestables *R-S*. En la figura 6.120*a* se representa el esquema de un biestable *R-S* de borrado prioritario sincronizado por niveles. La sincronización se realiza mediante sendas puertas *Y* de dos entradas conectadas a las entradas *R* y *S* del biestable, que se denominan R_s y S_s para indicar que están sincronizadas. Se conectan entre sí

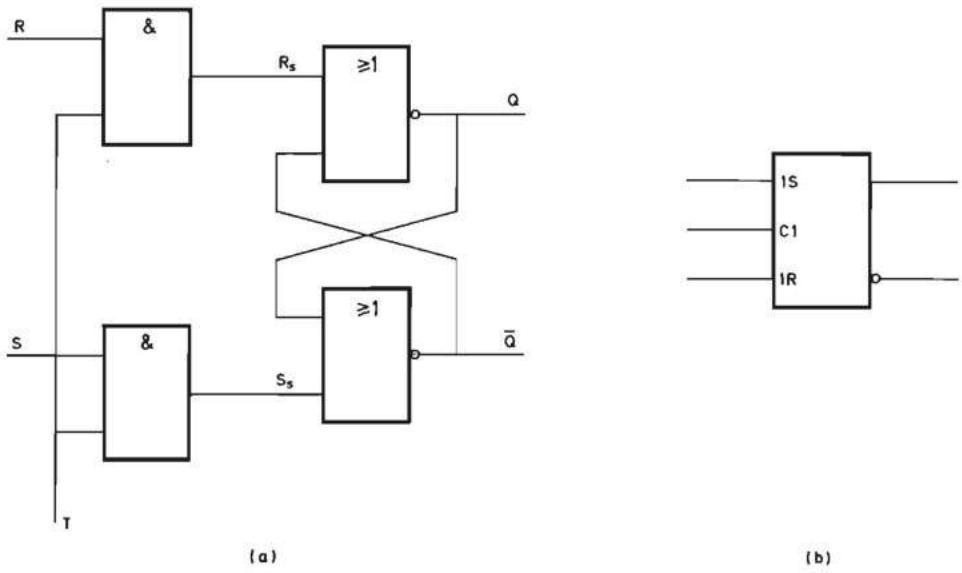


FIGURA 6.120.—**Biestable *R-S* de borrado prioritario sincronizado mediante el nivel uno de la entrada *T*: *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

sendas entradas de ambas puertas Y para obtener la entrada T de sincronización. La otra entrada de cada puerta Y se lleva al exterior y constituye la entrada R y S del biestable.

Solamente cuando la entrada T de sincronización se encuentra en el estado lógico uno, permite que las entradas R y S actúen sobre el estado del biestable. En la figura 6.120*b* se representa el símbolo normalizado de este biestable. La entrada de sincronización se indica con el símbolo C (control) seguido del dígito 1. Este mismo dígito se coloca antes de los símbolos S y R, para indicar que estas entradas están sincronizadas con la $C1$. [Al lector se le remite al apéndice 2 (apartado A2.5.3.2.) para estudiar la relación de control C.]

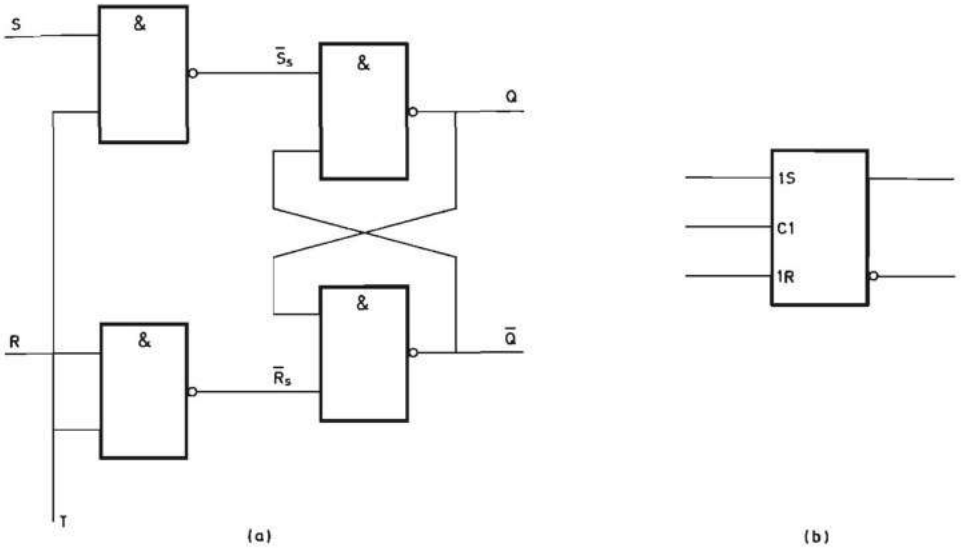


FIGURA 6.121.—**Biestable *R-S* de grabado prioritario sincronizado mediante el nivel uno de la entrada *T* (*C*1): *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

En la figura 6.120 se representa el biestable *R-S* sincronizado por niveles realizado con un biestable *R-S* de borrado prioritario. En la figura 6.121 se muestra su realización con un biestable de grabado prioritario [con puertas NO-Y (NAND)]. Al igual que en el primero, cuando la entrada T ($C1$) de sincronización se encuentra en nivel uno, permite que las entradas S ($1S$) y R ($1R$) actúen sobre el estado del biestable. Tanto en uno como en otro biestable se dice que la entrada T ($C1$) es activa con un nivel uno. Se obtiene un biestable *R-S* sincronizado por niveles en el que la entrada T ($C1$) es activa con un cero lógico añadiendo un inversor a las figuras 6.120 y 6.121. Como ejemplo se representa la figura 6.122 obtenida a partir de la 6.121.

El más utilizado de los biestables sincronizados por niveles es el tipo D que recibe el nombre de biestable D cerrojo (latch).

En la figura 6.123 se indica el esquema lógico de un biestable típico de este tipo. Mientras la señal en T está en estado lógico uno, la información presente en

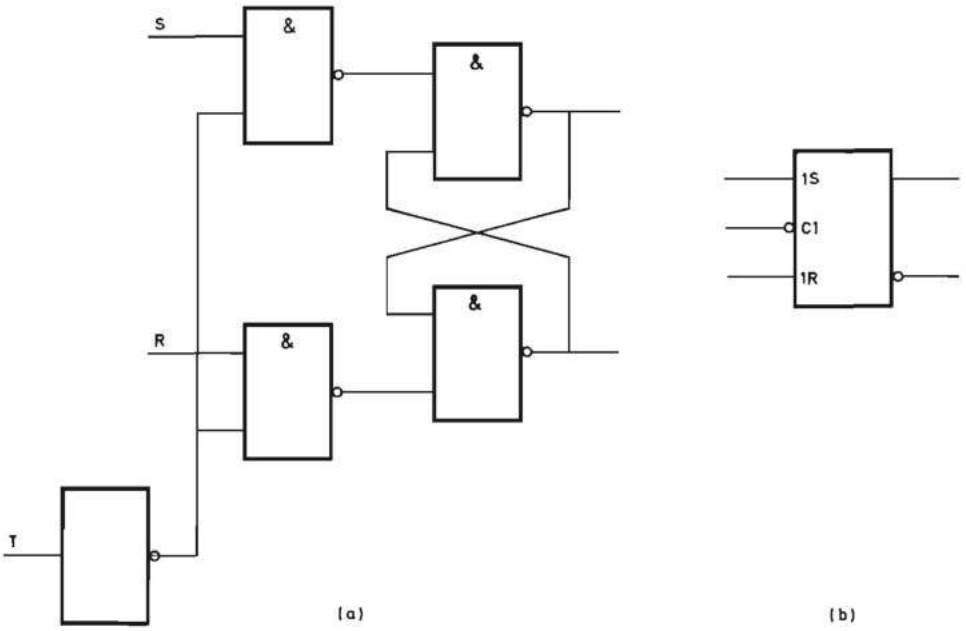


FIGURA 6.122.—**Biestable *R-S* de grabado prioritario sincronizado mediante el nivel cero de la entrada *T*(*C*1): *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

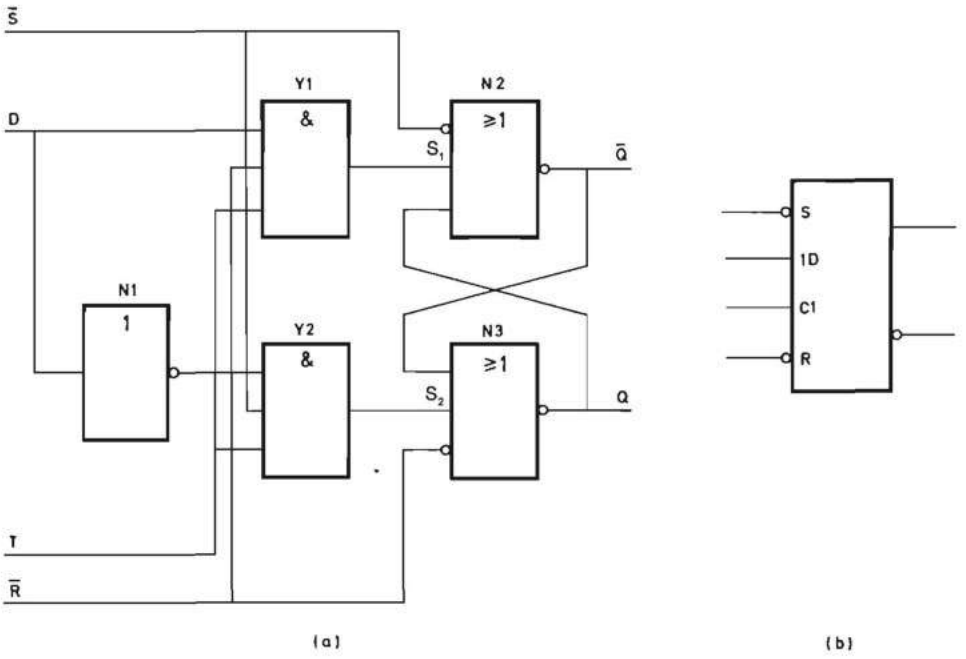


FIGURA 6.123.—**Biestable *D* cerrojo (latch) sincronizado por el nivel uno de la entrada *T*: *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

la entrada D y su inversa (obtenida mediante N_1) se aplican a través de Y_1 e Y_2 respectivamente, a las entradas de puesta a uno (S_1) y puesta a cero (S_2) del biestable formado por N_2 y N_3. Por lo tanto, la salida del biestable sigue las variaciones de la información presente en la entrada D y el conjunto se comporta como una puerta Y. Es importante notar la diferencia de funcionamiento con el biestable D activado por flancos que se describe en el apartado siguiente.

Tan pronto como la señal en T toma de nuevo el valor lógico cero, las dos entradas S_1 y S_2 del biestable pasan al estado lógico cero y en el biestable queda almacenada la información presente en D en el instante en que se produce la bajada a cero de la señal T. Mientras esta señal permanece en estado cero, el estado del biestable se mantiene invariable, independientemente de las variaciones que se produzcan en la información presente en la entrada D. En la figura 6.123*b* se representa el símbolo lógico normalizado correspondiente a este biestable.

Para que un biestable sincronizado memorice adecuadamente la información presente en las entradas correspondientes cuando la entrada T adopta un nivel activo, es necesaria una adecuada relación de tiempo entre ambas.

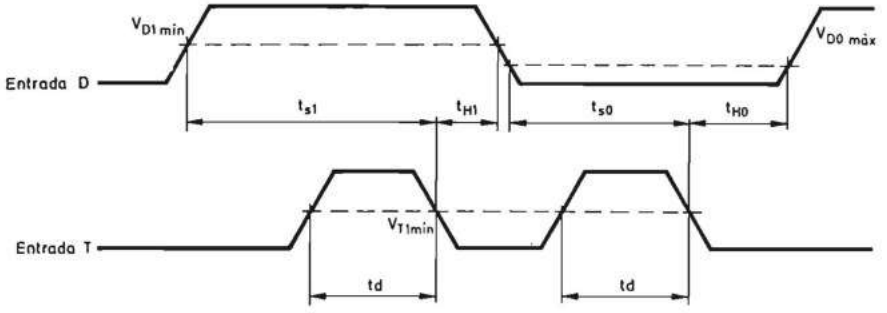


FIGURA 6.124.—**Relación temporal entre las señales aplicadas a las entradas D y T de un biestable cerrojo (latch) sincronizado mediante el nivel uno de la entrada T ($C1$).**

En la figura 6.124 se representa dicha relación entre las entradas T ($C1$) y D ($1D$) de un biestable cerrojo (latch) cuya entrada T ($C1$) es activa con el nivel uno. Las especificaciones que ambas entradas deben cumplir son las siguientes:

1. La información presente en la entrada D ha de permanecer en estado cero o en estado uno durante un tiempo mínimo t_{s0} o t_{s1} antes de la aparición del flanco de bajada del estado lógico uno al cero de la señal en T (fig. 6.124).
2. Además dicha información ha de permanecer invariable un tiempo mínimo t_{H0} o t_{H1} para los estados lógicos cero y uno respectivamente después de haberse producido el flanco de bajada de la señal en T ($C1$) (fig. 6.124).
3. La señal aplicada a la entrada T ($C1$) debe permanecer en el nivel activo (en este caso el nivel uno lógico) durante un tiempo mínimo t_d (fig. 6.124). Los valores de t_{s0}, t_{s1}, t_{H0}, t_{H1} y t_d dependen de la tecnología utilizada en la realización del biestable y limitan la máxima frecuencia de los impulsos que se pueden aplicar a la entrada T ($C1$).

En la figura 6.123 se observa que además de las entradas D y T, el biestable posee entradas de puesta a cero (R) y puesta a uno (S) asíncronas que actúan por nivel y que en esta ocasión son activas con un cero lógico. Estas entradas permiten poner a cero o a uno el biestable independientemente de la entrada de sincronización T y la entrada de información D. Se utilizan usualmente para colocar en estado inicial los biestables en el momento de dar tensión al equipo del cual forman parte.

Este biestable se representa mediante un diagrama de bloques o símbolo lógico. En la figura 6.125 se representan los símbolos lógicos no normalizado (a) y normalizado (b). En el símbolo no normalizado, las entradas D y T se colocan en el segmento vertical izquierdo y las salidas Q y $\overline{Q}$ en el vertical derecho. Las entradas asíncronas por nivel S y R se indican en los segmentos horizontales; la de puesta a uno ($\overline{S}$) en la parte superior y la de puesta a cero ($\overline{R}$) en la inferior. En el símbolo normalizado todas las entradas se colocan en el segmento vertical izquierdo, y las salidas en el derecho. La entrada de sincronización se denomina $C1$, la de información sincronizada con ella $1D$ y las de puesta a uno y puesta a cero S y R respectivamente.

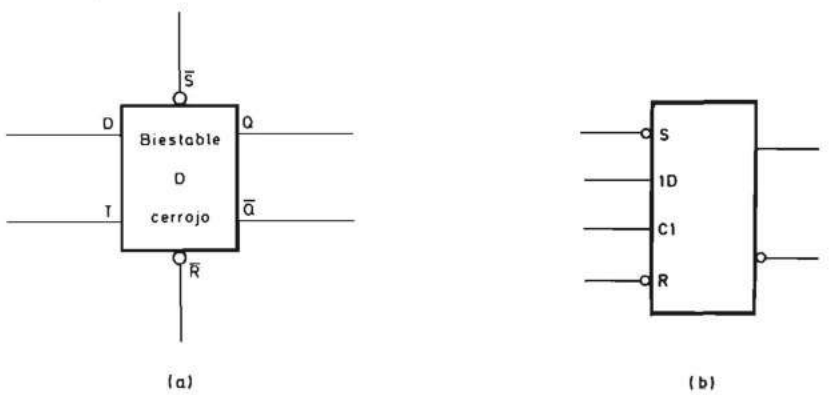


FIGURA 6.125.—**Símbolos lógicos del biestable D cerrojo de la figura 6.123: *a*) No normalizado; *b*) Normalizado**

Este biestable se ha utilizado en la realización de registros auxiliares para almacenamiento temporal de la información. Estos registros, cuyas aplicaciones se estudian en el apartado 6.3.5.4.5, están constituidos por un cierto número de biestables D cerrojo, cuyas entradas T se unen entre sí y las salidas Q y entradas D están disponibles al exterior independientemente.

6.3.2.2 Biestables sincronizados por cambios de nivel. Estos biestables se realizan también con biestables R-S activados por niveles. Existen dos formas diferentes de interconectar entre sí los biestables R-S que dan lugar a dos grandes clases de biestables sincronizados por cambios de nivel:

a) Biestables activados por impulsos (pulse-triggered), en los que la información se memoriza durante un nivel activo y se transfiere a la salida en el flanco que le sigue que se denomina también activo. Estos biestables presentan a su vez dos variantes que se estudian en el próximo apartado.
b) Biestables activados por flancos (edge-triggered) en los que tanto la memorización de la información en el biestable como la aparición de la misma a la salida se realiza en un cierto flanco que se denomina activo.

Tanto en los biestables activados por impulsos como en los activados por flancos, el flanco activo puede ser el de subida o el de bajada. En sucesivos apartados se analizan ambas clases de biestables.

6.3.2.2.1 Biestables activados por impulsos. Se clasifican a su vez en dos tipos:

a) Biestables principal-subordinado (master-slave).
b) Biestables principal-subordinado con enclavamiento (master-slave with data lockout).

A continuación se analiza cada uno de ellos.

6.3.2.2.1.1 Biestables principal-subordinado (master-slave). El diagrama de bloques de un biestable de este tipo se representa en la figura 6.126. Está constituido por dos biestables sincronizados por niveles interconectados entre sí. El primero (master) recibe las entradas de información y su salida se conecta a la entrada del segundo (slave) cuya salida constituye la salida del conjunto. Las entradas T (C) de ambos biestables son mutuamente inversas. Analicemos el comportamiento de este sistema. Se supone que la entrada T de cada biestable permite que las entradas de información actúen sobre su estado cuando se encuentra en estado lógico uno. Mientras la entrada T se encuentra en estado cero, las entradas de información no actúan sobre el primer biestable, pero, por el contrario, las salidas de éste actúan sobre el estado del segundo. Este tiene el mismo estado que aquél y, por ello se le llama subordinado (slave). Cuando la entrada de sincronización pasa

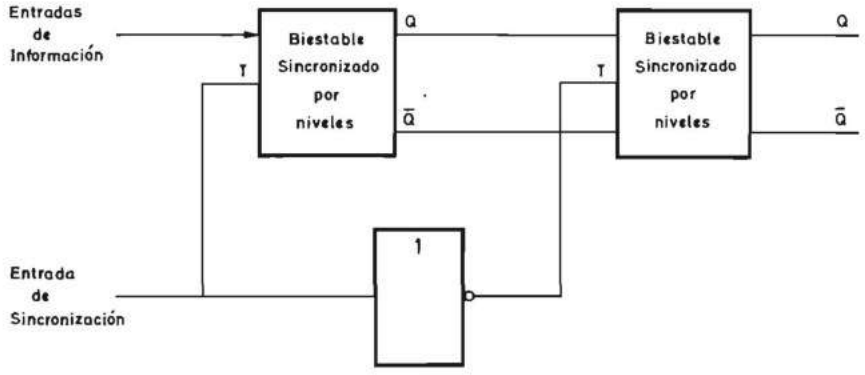


FIGURA 6.126.—**Biestable principal-subordinado (master-slave).**

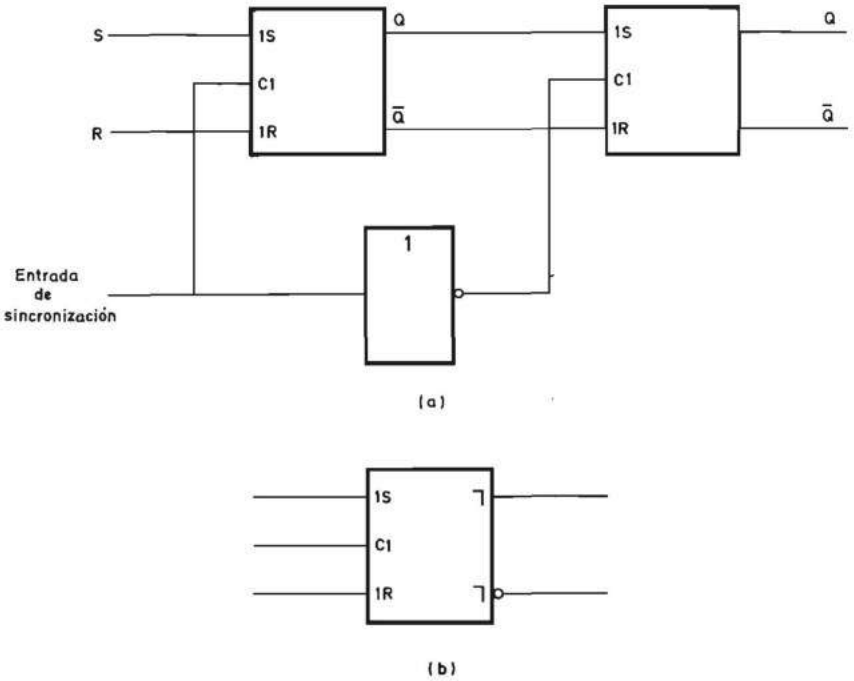


FIGURA 6.127.—**Biestable *R-S* principal-subordinado activado por impulsos positivos (nivel uno): *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

al estado uno, la información presente en las entradas de información actúa sobre el primer biestable y simultáneamente el subordinado queda desconectado de él. Cuando la entrada de sincronización vuelve al estado cero, el biestable subordinado vuelve a adoptar el estado del principal.

El lector puede observar que este biestable no es totalmente activado por flancos porque la información entra en el principal (master) mientras la entrada de sincronización T se encuentra en estado uno. Por el contrario, la salida de la información Q se produce por flanco porque la información aparece en ella cuando la entrada T cambia de uno a cero.

Los biestables principal-subordinado (master-slave) pueden ser *R-S*, *J-K* o *D*. En la figura 6.127*a* se representa el biestable *R-S* principal-subordinado formado por dos biestables *R-S* activados por niveles unidos entre sí. En la figura 6.127*b* se representa el símbolo lógico normalizado. En la entrada *C*1 no se coloca el símbolo de flanco debido a que la información se almacena durante el nivel activo (en este caso el uno) de la entrada *C*1 del primer biestable. Pero dado que la información aparece a la salida después del flanco activo (en este caso de bajada) de *C*1, se coloca el símbolo ⌝ en ella para distinguir este biestable del sincronizado por niveles. El biestable *J-K* se obtiene realimentando las salidas Q y $\overline{Q}$ a través de puertas Y tal como se indica en la figura 6.128*a*. La ventaja de este biestable es que permite que ambas entradas J y K adopten el estado uno simultáneamente

y en esta situación el impulso aplicado a la entrada T hace cambiar de estado la salida Q. En la figura 6.128*b* se representa el símbolo lógico de este biestable cuya entrada $C1$ transfiere también la información al biestable de salida al pasar de uno a cero.

En la figura 6.129*a* se representa el esquema de un biestable *J-K* principal-subordinado. Cuando se aplica un impulso a la entrada T de sincronización, el estado adoptado por el biestable es el que se indica en su tabla de excitación. Esta tabla, que se representa en la tabla 6.16, coincide con la del biestable *J-K* no sin-

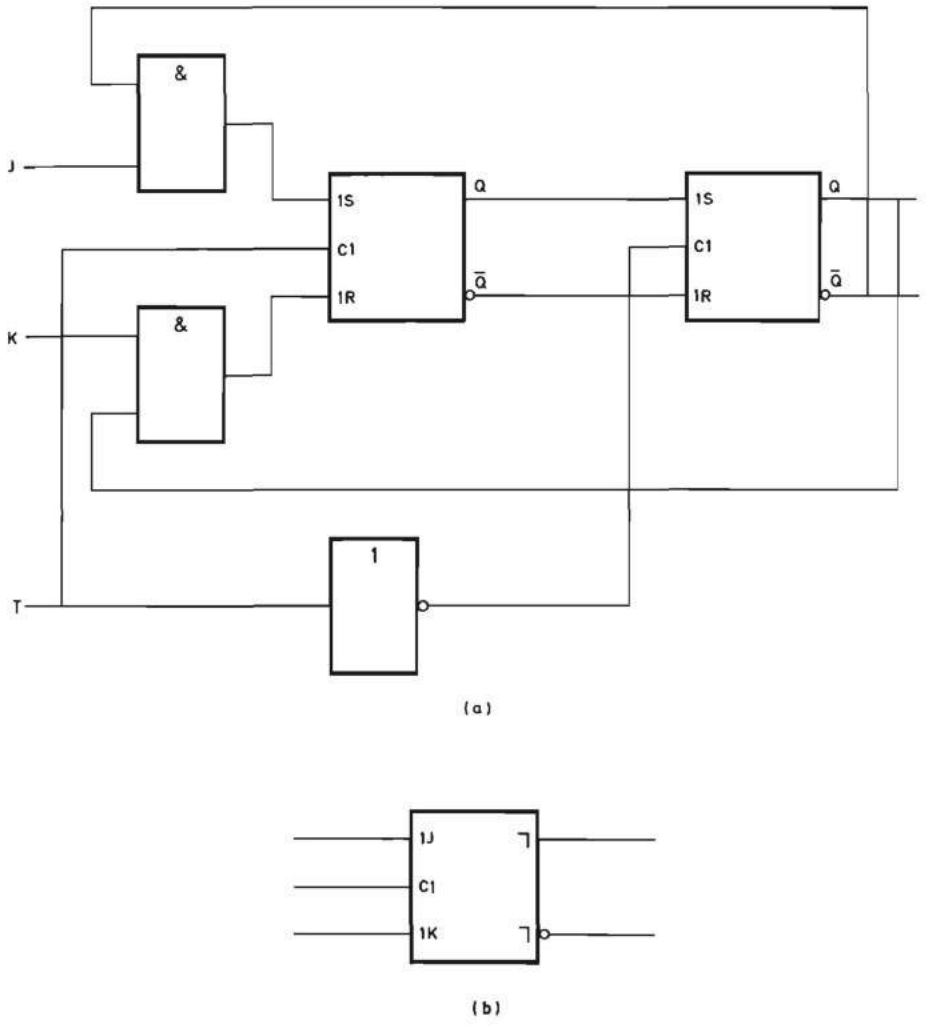


FIGURA 6.128.—**Biestable *J-K* principal-subordinado activado por impulsos positivos (nivel uno): *a*) Esquema simplificado del circuito; *b*) Símbolo lógico normalizado.**

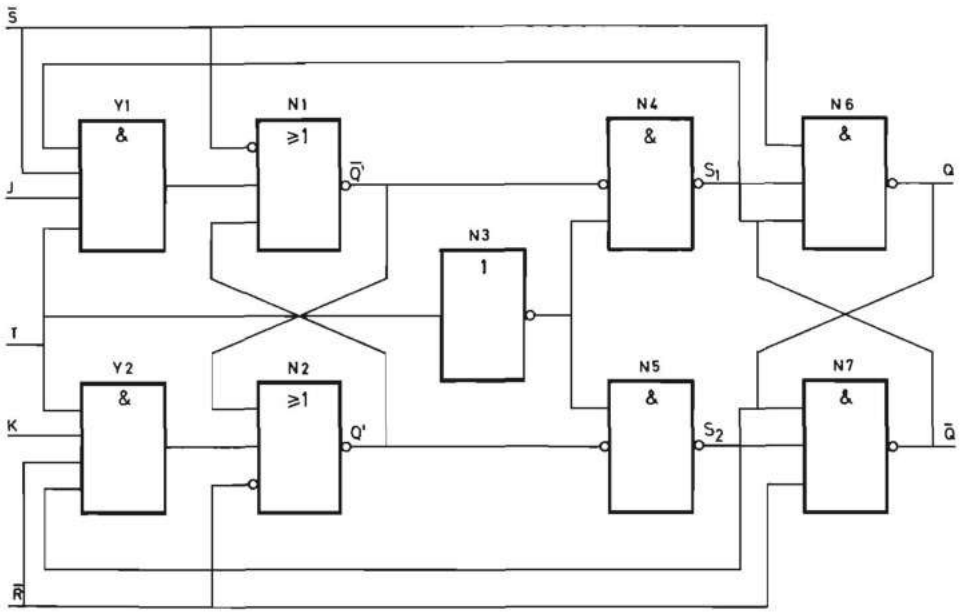


FIGURA 6.129.—**Esquema del circuito de un biestable *J-K* principal-subordinado, activado por impulsos positivos (nivel uno).**

J	K	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

(a)

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	$\bar{Q}_t$

(b)

TABLA 6.16.—**Tablas de excitación del biestable *J-K*: *a*) Completa; *b*) Reducida.**

cronizado (tablas 6.5 y 6.6). (La variable de estado interno se denomina Q en lugar de Y, siguiendo la nomenclatura normalmente utilizada por los fabricantes de circuitos integrados.) De esta tabla se deduce la ecuación característica del biestable. Para ello se representa la tabla de excitación en una tabla de Karnaugh (fig. 6.130) y se agrupan todos los unos correspondientes a la fila $Q_1 = 0$ y los de la fila $Q_1 = 1$ con lo cual se obtiene la ecuación característica:

$$Q_{t+1} = J\overline{Q}_t + \overline{K}Q_t$$

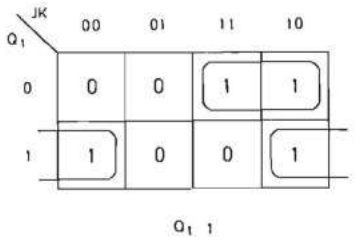


FIGURA 6.130.—**Tabla de excitación de un biestable *J-K* principal-subordinado.**

que indica el nuevo estado de Q después de aplicar un impulso en T en función del estado anterior y del estado lógico de J y K.

A continuación se describe la forma de operación de este biestable.

Cuando la señal en T pasa del estado lógico cero al uno, la información presente en las entradas de las puertas Y_1 e Y_2 (fig. 6.129) se transfiere a la salida de las mismas y activa al biestable formado por las puertas N_1 y N_2 que recibe el nombre de principal (master).

Las puertas Y_1 e Y_2 realizan respectivamente (mientras T está en estado uno) el producto de la información externa presente en las entradas J ($1J$) y K ($1K$) por la que en ese instante tiene almacenada el biestable de salida subordinado (slave) formado por las puertas N_6 y N_7. De esta forma se consigue que, al producirse el flanco de subida de la señal en T ($C1$), se almacene en el biestable principal (master) la información correspondiente a la tabla 6.17, de acuerdo con el valor de la información presente en las entradas J y K y el estado actual del biestable subordinado (slave).

Cuando la señal en T pase al estado lógico cero, la salida N_3 sube al estado lógico uno. De esta forma, a través de N_4 y N_5, la información presente en el biestable principal (master) se aplica a las entradas de puesta a uno (S_1) y puesta a cero (S_2) del biestable subordinado (slave) a través de N_4 y N_5 respectivamente, y se logra que el biestable subordinado tome el estado correspondiente a la tabla de excitación (tabla 6.16). Simultáneamente se cierran las puertas Y_1 e Y_2 impi-

J	K	Q_t	Q'
0	0	X	Q'
0	1	0	Q'
0	1	1	1
1	0	0	0
1	0	1	Q'
1	1	0	0
1	1	1	1

TABLA 6.17

diendo el cambio del biestable principal mientras la entrada T permanece en estado lógico cero.

Si se desea que el biestable principal cambie de estado únicamente cuando la señal en T pase del estado lógico cero al uno, es necesario que las entradas J y K permanezcan invariables durante todo el tiempo que la entrada T se encuentre en estado uno.

Aunque el nivel de la señal en T es el que controla la entrada de información en el biestable principal, la información pasa al subordinado (slave) en el flanco de bajada de la señal en T y, por lo tanto, este biestable se puede considerar activado por flancos si se cumple la condición anteriormente indicada.

En el estudio realizado, la información entraba en el biestable principal en el nivel uno de la entrada T y pasaba al subordinado en el cambio de uno a cero. Invirtiendo la situación del inversor N_3 de la figura 6.129 se puede cambiar la forma de operación del biestable. En efecto, si la entrada T se conecta directamente a N_4 y N_5 y la entrada de N_3 y la salida de éste se conecta a Y_1 e Y_2, se logra que la información entre en el biestable principal cuando el nivel de la entrada T es un cero lógico y pase al subordinado cuando cambie de cero a uno.

Una propiedad importante de este biestable es que funciona en los cambios de nivel de la señal presente en la entrada T sin que la pendiente de los flancos, tanto de bajada como de subida, tenga efecto alguno.

Al igual que el biestable sincronizado por niveles estudiado en el apartado anterior, los biestables principal-subordinado poseen entradas de puesta a cero y puesta a uno asíncronas activadas por niveles, en las que el nivel activo puede ser un cero o un uno lógicos. En la figura 6.129 se observa que, además de las entradas J y K sincronizadas mediante la entrada T, existen entradas no sincronizadas de puesta a uno (S) y puesta a cero (R) que permiten poner a uno y cero, respectivamente, ambos biestables principal y subordinado independientemente del estado lógico del resto de las entradas. Esto se realiza aplicando un cero lógico a una de dichas entradas.

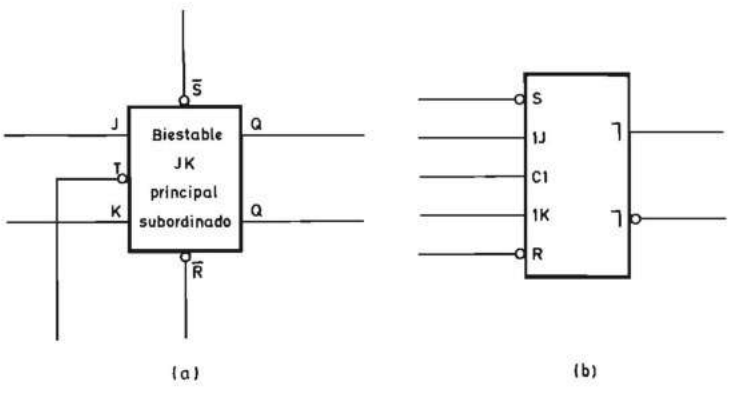


FIGURA 6.131.—**Símbolos lógicos de un biestable *J-K* principal-subordinado activado por impulsos positivos: *a*) No normalizado; *b*) Normalizado.**

En la figura 6.131 se representan los símbolos lógicos no normalizado (fig. 6.131*a*) y normalizado (fig. 6.131*b*) correspondientes al biestable principal-subordinado de la figura 6.129.

Las señales presentes en las entradas de información de un biestable principal-subordinado y la entrada de sincronización T han de cumplir también determinadas relaciones en el tiempo. En la figura 6.132 se representan dichas relaciones en el supuesto de que la información entra en el principal (master) en el nivel uno de la señal aplicada a la entrada T y pasa al subordinado (slave) en el flanco de bajada (paso de cero a uno) de dicha señal.

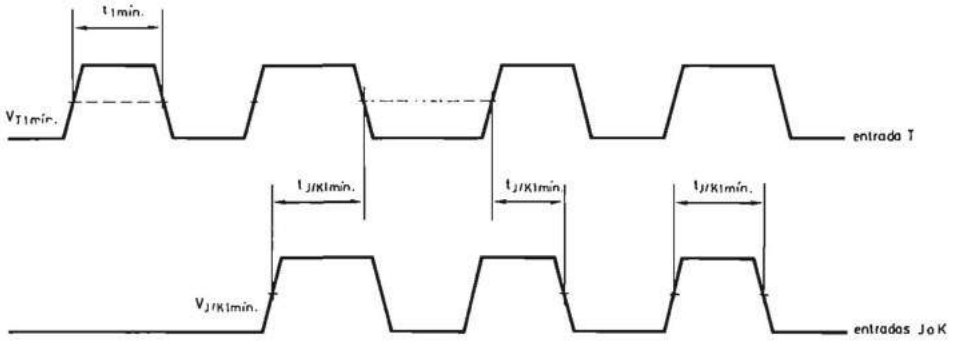


FIGURA 6.132.—**Relación temporal entre las señales de un biestable *J-K* principal-subordinado activado con impulsos positivos.**

Las especificaciones más importantes para una operación correcta son:

a) La señal presente en T ha de permanecer en estado lógico uno un tiempo mínimo $t_{T1\ min}$.

b) La información en J o K ha de permanecer en estado uno mientras está en dicho estado la señal en T durante un tiempo mínimo $t_{J/K1\ min}$.

En algunas tecnologías se han realizado biestables D principal-subordinado. Dichos biestables están formados por un biestable D sincronizado por niveles como el representado en las figuras 6.123 y 6.125 seguido de un biestable *R-S* sincronizado también por niveles (figs. 6.120 y 6.121). Su diagrama de bloques se representa en la figura 6.133, en la que el biestable memoriza la información en el principal cuando la entrada T se encuentra en nivel cero y la transfiere al subordinado en el cambio de cero a uno (flanco de subida) de dicha señal. El lector no tendrá ninguna dificultad para comprender su funcionamiento a la luz de la teoría expuesta en párrafos anteriores. Es conveniente resaltar que el biestable D principal-subordinado no presenta el inconveniente del *J-K* que necesitaba que ambas entradas no variasen mientras la entrada T se encontraba en nivel activo, si se quería asegurar que lo que se transfería al subordinado en el flanco activo era el estado correspondiente a la situación de aquéllas en dicho instante. En efecto, si la entrada D varía mientras T se encuentra en el nivel activo (nivel cero en la figu-

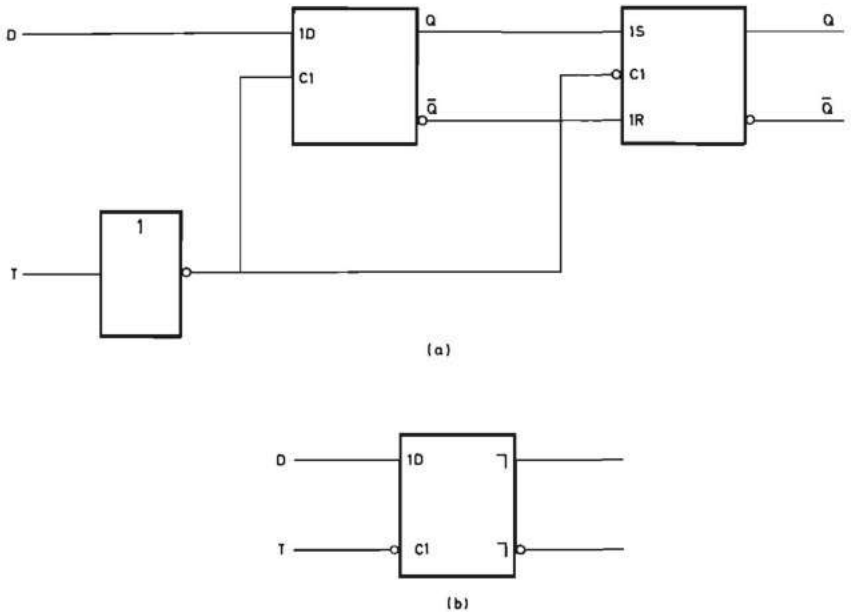


FIGURA 6.133.—**Biestable *D* principal-subordinado activado por impulsos negativos y símbolo lógico normalizado.**

ra 6.133*a*), la salida *Q* del biestable principal irá cambiando de acuerdo con ella. El estado de la entrada *D* en el flanco activo (en este caso el de subida) será el que se transferirá a la salida. Por lo tanto, el biestable *D* principal-subordinado es, en realidad, totalmente activado por flancos y su símbolo normalizado es el representado en la figura 6.133*b*.

6.3.2.2.1.2 Biestables principal-subordinado con enclavamiento (master-slave with data lockout). En el apartado anterior se analizaron los biestables principal-subordinado y se comprobó que presentan el inconveniente de que las entradas de información han de permanecer fijas durante todo el tiempo que la entrada de sincronización o control se encuentra en el nivel que hace que el estado de aquéllas se introduzca en el biestable principal. Para superar dicho inconveniente se crearon los biestables principal-subordinado con enclavamiento (master-slave with data lockout). En la figura 6.134*a* se representa el esquema de un biestable *R-S* de este tipo. La diferencia fundamental con el estudiado en el apartado anterior es la introducción de un nuevo biestable formado por *N*5 y *N*6 cuyas salidas se realimentan a las entradas del biestable principal (master) formado por *N*3 y *N*4. De esta forma se logra que, al iniciarse el impulso (flanco de subida en la figura 6.134*a*), la información entre en el biestable principal y que se transfiera al subordinado (slave) formado por *N*9 y *N*10 al pasar la entrada *T* de uno a cero. El lector interesado puede analizar el esquema de la figura 6.134*a* y comprobar el funcionamiento descrito.

En la figura 6.134*b* se representa el símbolo lógico normalizado de este biestable que se diferencia del biestable principal-subordinado (figura 6.127) en que la entrada *C* posee el indicativo del flanco para especificar que la información sólo tiene que permanecer estable al producirse el flanco inicial.

En la figura 6.135*a* se representa un biestable *J-K* principal-subordinado con enclavamiento en el que se introduce la información en el principal (*N*3-*N*4) en el flanco de subida y pasa al subordinado (*N*11-*N*12) en el de bajada, lo cual se indica en el símbolo normalizado correspondiente (figura 6.135*b*).

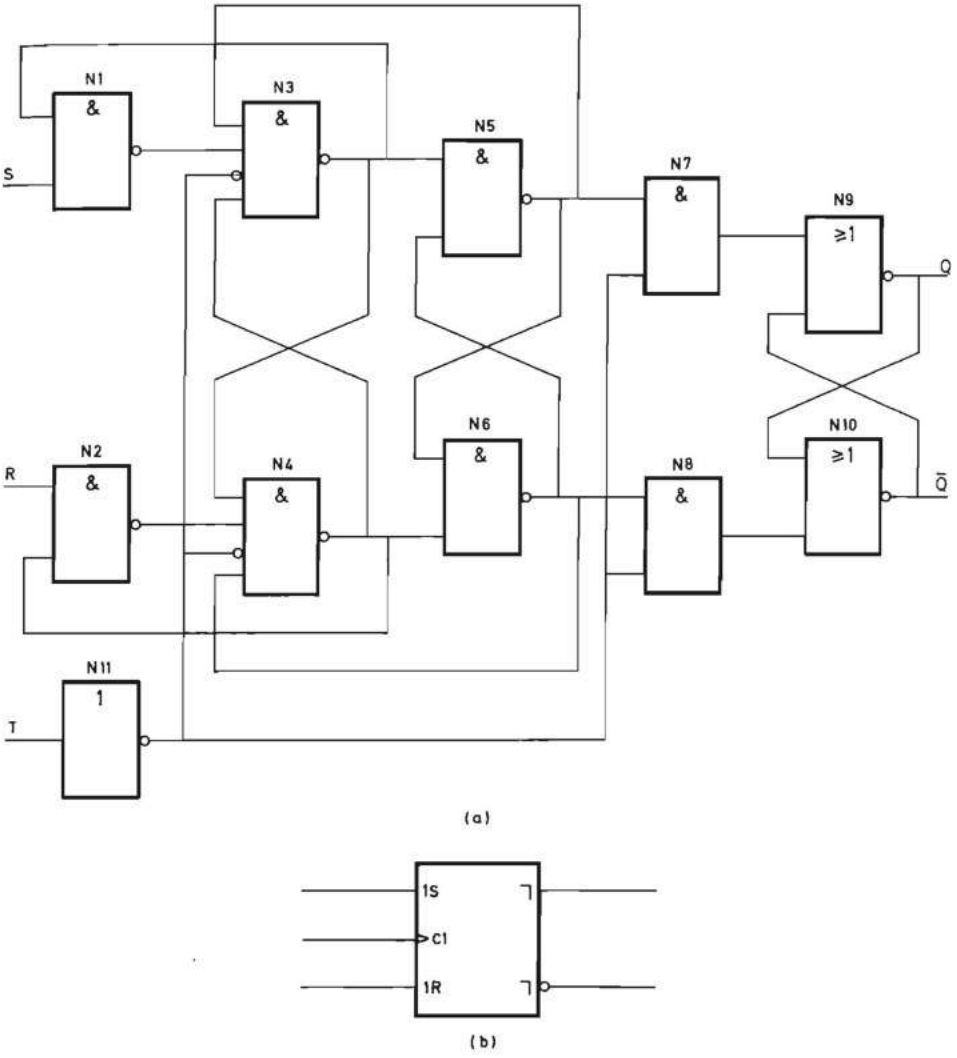


FIGURA 6.134.—**Biestable *R-S* principal-subordinado con enclavamiento, activado mediante impulsos positivos: *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

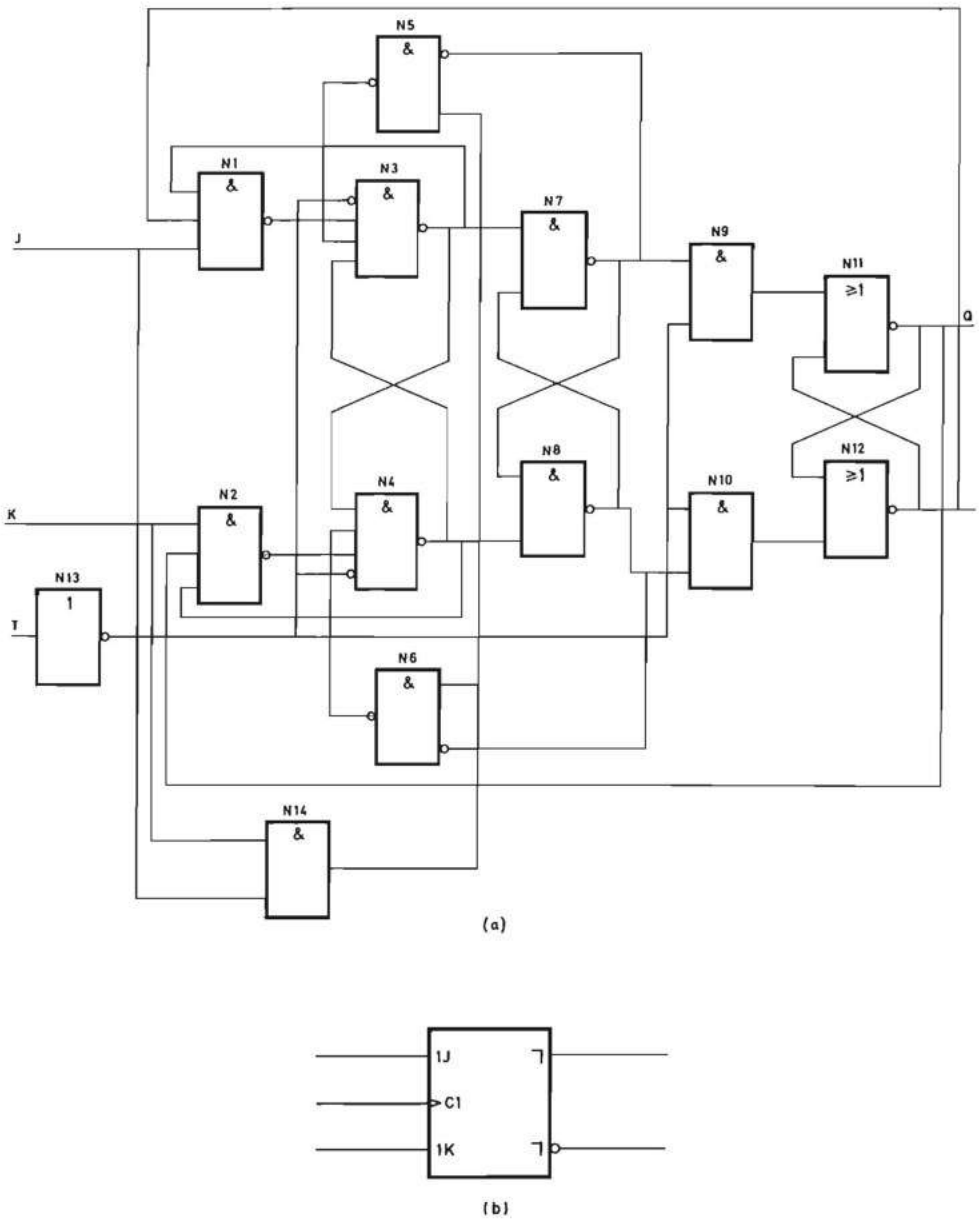


FIGURA 6.135.—**Biestable *J-K* principal subordinado con enclavamiento activado, mediante impulsos positivos: *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

6.3.2.2.2 Biestables activados por flancos (edge-triggered). Reciben este nombre los biestables en los que la información se transfiere a la salida en el mismo flanco en que se memoriza, lo que los distingue tanto de los principal-subordi-

nado (master-slave) como de los principal-subordinado con enclavamiento (master-slave with data lockout).

Tal como se estudió anteriormente, el biestable *D* principal-subordinado se comporta realmente con activación total por flancos. Pero existe otra forma de realizar los biestables de activación por flancos diferente a la del montaje principal-subordinado que permite la obtención de biestables *J-K* y *D* totalmente activados por flancos. En esta nueva forma se utilizan tres biestables activados por niveles adecuadamente conectados entre sí.

En la figura 6.136 se representa el esquema del biestable *D* activado por flancos cuya tabla de excitación se indica en la tabla 6.18. En dicha figura se comprueba que este biestable está constituido a su vez por tres biestables *R-S* formados por las puertas *N*1-*N*2, *N*3-*N*4 y *N*5-*N*6.

El biestable de la figura 6.136 posee entradas de puesta a uno (*S*) y puesta a cero (*R*) asíncronas que actúan sobre el biestable cuando se encuentran en el nivel

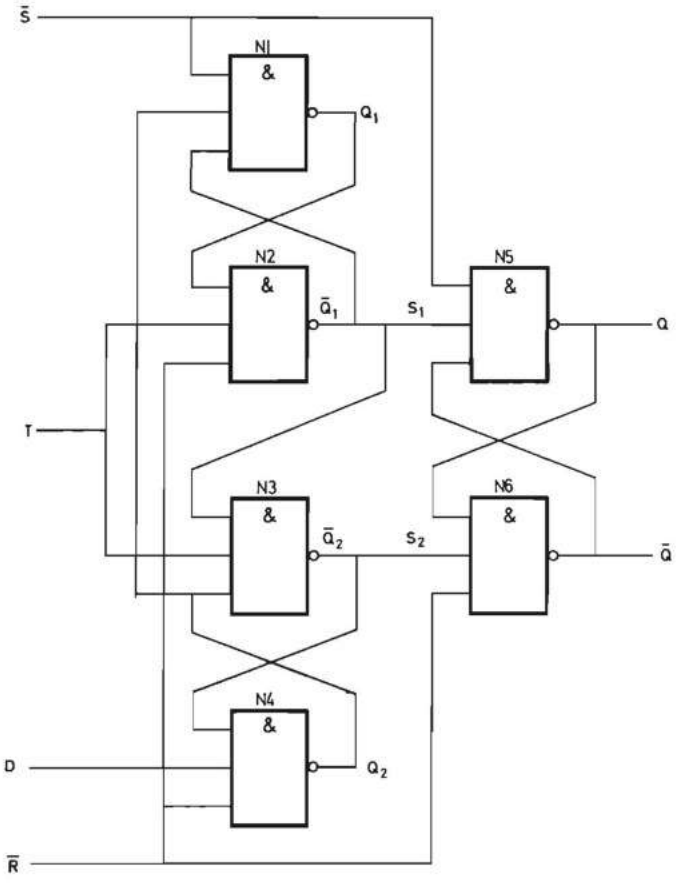


FIGURA 6.136.—**Esquema de un biestable *D* activado por flancos de subida.**

D	Q_t	Q_{t+1}
0	0	0
0	1	0
1	0	1
1	1	1

a)

D	Q_{t+1}
0	0
1	1

b)

TABLA 6.18.— **Tablas de excitación de un biestable D activado por flancos: *a*) Completa; *c*) Reducida.**

activo cero y realizan dicha acción sobre las salidas Q (directa) y $\overline{Q}$ (inversa) del biestable.

Es interesante analizar seguidamente el funcionamiento de las entradas T y D de este biestable cuando S y R se encuentran en nivel uno. Mientras la señal T está en estado lógico cero, las entradas de puesta a uno S_1 y puesta a cero S_2 del biestable de salida formado por las puertas N_5 y N_6 están en estado lógico uno y, por tanto, el estado de éste permanece invariable. Al mismo tiempo, en las salidas Q_1 y Q_2 de los otros dos biestables aparecen, respectivamente, el estado lógico correspondiente a la entrada D y el inverso del mismo. Cuando T pasa del estado lógico cero al uno, aparece un cero lógico en S_1 o S_2 según esté en estado lógico uno Q_1 o Q_2, respectivamente. De esta forma se transfiere a las salidas Q y $\overline{Q}$ el estado lógico de D y su inverso, respectivamente.

En este biestable la transferencia de información se produce en el instante en que la entrada T cambia del estado cero al uno sin que influya la velocidad de dicho cambio.

Las señales aplicadas a las entradas D y T de este biestable también han de cumplir unas ciertas especificaciones para asegurar la correcta memorización de la información. Dichas relaciones, que se representan en la figura 6.137, son las siguientes:

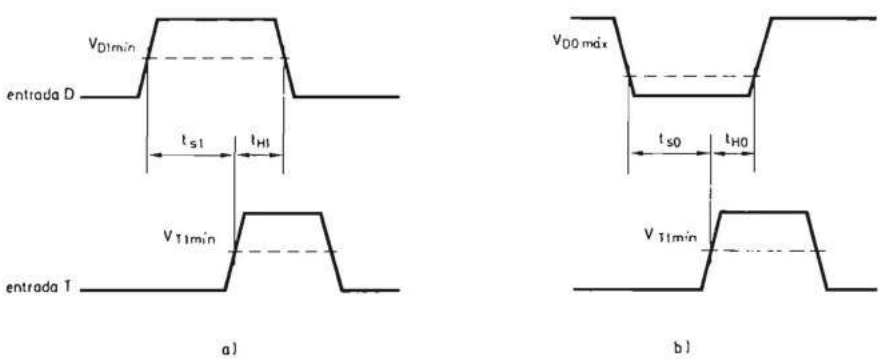


FIGURA 6.137.—**Relación temporal entre las señales del biestable D activado por flancos de la figura 6.136.**

1. La información presente en la entrada D ha de permanecer en estado cero o uno durante un tiempo mínimo t_{s0} o t_{s1}, respectivamente, antes de la aparición del flanco de subida del estado lógico cero al uno de la señal en T.
2. La información presente en la entrada D ha de permanecer en estado cero o uno durante un tiempo mínimo t_{H1} o t_{H0}, respectivamente, después de la aparición del flanco de subida del estado lógico cero al uno de la señal en T.

En la figura 6.138 se representan los símbolos lógicos no normalizado (figura 6.138*a*) y normalizado (figura 6.138*b*) correspondientes a este biestable. En este último es conveniente destacar que no posee indicativo alguno en las salidas y que la entrada C ($C1$) posee el indicativo de flanco.

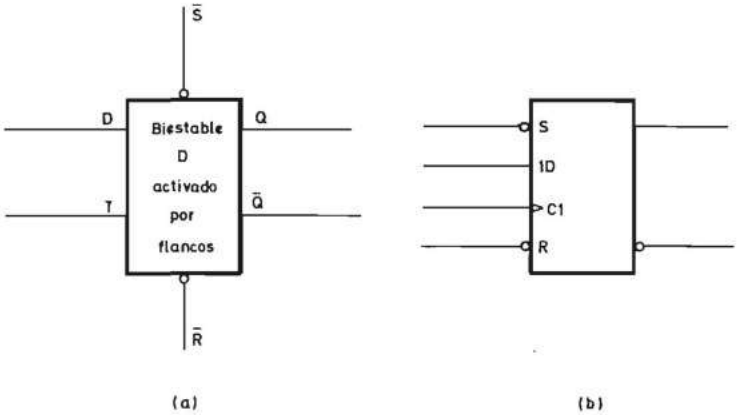


FIGURA 6.138.—**Símbolos lógicos del biestable D activado por flancos de subida cuyo esquema se representa en la figura 6.136: *a*) No normalizado; *b*) Normalizado.**

En la figura 6.139*a* se representa el esquema de un biestable J-$\overline{K}$ de activación por flancos de subida cuyo análisis se propone al lector. La entrada K está invertida lo cual indica que se memoriza un uno cuando $J = \overline{K} = 1$ y un cero cuando $J = \overline{K} = 0$.

En las figuras 6.139*b* y 6.139*c* se representan los símbolos lógicos no normalizado y normalizado correspondientes respectivamente.

6.3.3 Aplicación de los biestables sincronizados activados por flancos (edge-triggered) a la síntesis de biestables asíncronos activados por flancos

En el apartado anterior se estudiaron los biestables síncronos activados por flancos. Como el lector recordará, estos biestables poseen dos tipos de entradas: las de información, que determinan el estado del biestable, y la de sincronismo, que controla el instante en que se produce dicha acción.

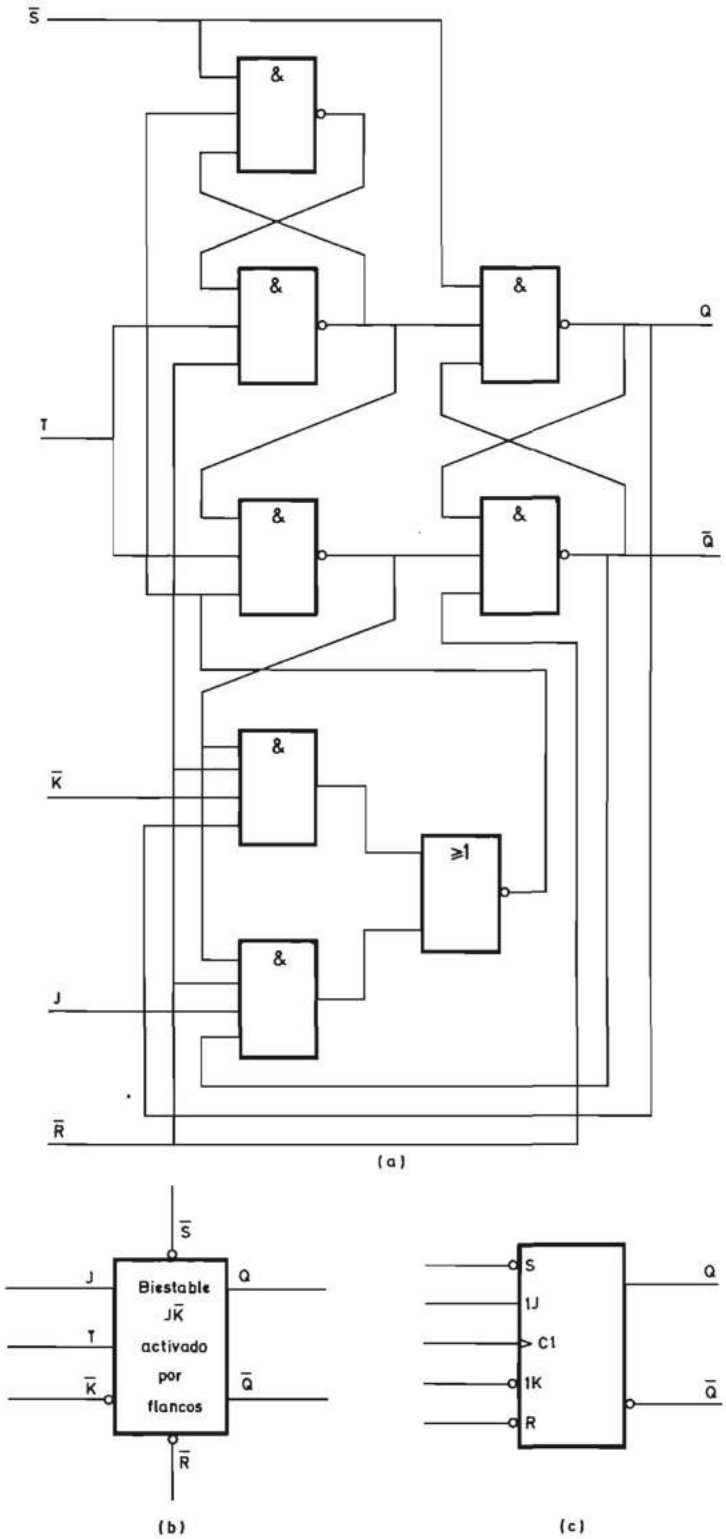


FIGURA 6.139.— **Biestable *J-K* activado por flancos (edge-triggered) de subida: *a*) Esquema del circuito; *b*) Símbolo lógico no normalizado; *c*) Símbolo lógico normalizado.**

Pero la operación lógica producto del cambio de estado de una variable por el estado de otras permite analizar los biestables síncronos desde un nuevo punto de vista. Las ecuaciones obtenidas para los biestables *D* y *J-K* activados por flancos se pueden representar de una nueva forma:

Biestable *D*

$$Q_{t+1} = D \Leftrightarrow Q = T \uparrow (D)$$

Biestable *J-K*

$$Q_{t+1} = J\overline{Q}_1 + \overline{K}Q_1 \Leftrightarrow Q = T \uparrow (J\overline{Q} + \overline{K}Q)$$

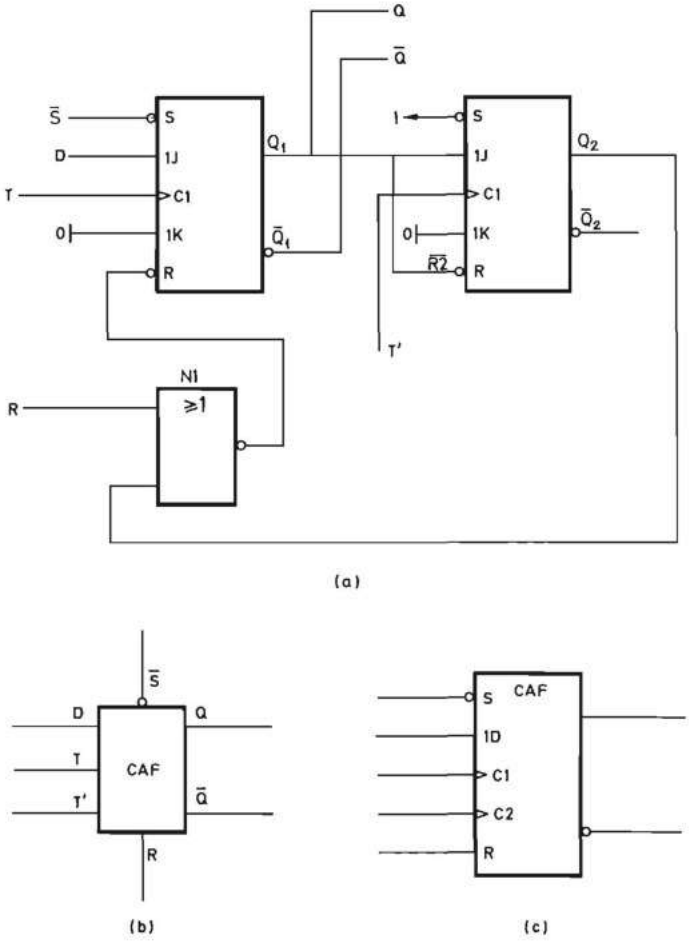


FIGURA 6.140.—**Célula CAF realizada con biestables *J-K* activados por flancos: *a*) Esquema del circuito; *b*) Símbolo lógico no normalizado; *c*) Símbolo lógico normalizado.**

En estas expresiones se supone que las entradas T actúan mediante los flancos de subida. Si los flancos activos fuesen los de bajada, se sustituiría el símbolo ↑ por el ↓.

La conexión adecuada de biestables *J-K* activados por flancos o biestables *D* activados por flancos permite la realización de células asíncronas activadas por flancos (CAF) que, como se vio en el apartado 6.2.3.4, pueden ser utilizadas en la síntesis de autómatas asíncronos de pequeña complejidad.

En la figura 6.140*a* se representa el esquema de la célula CAF realizada con biestables *J-K* activados por flancos. El biestable Q_1 tiene su entrada K conectada al nivel del cero lógico y sus entradas T y J constituyen respectivamente las entradas T y D de la célula; Q_1 sube al estado uno cuando se aplica un flanco activado a T y simultáneamente D se encuentra en estado uno. Por lo tanto, Q_1 es la salida Q de la célula. El biestable Q_2 permite el borrado de la célula mediante un flanco activo aplicado en su entrada T que constituye la entrada T'. Mientras Q_1

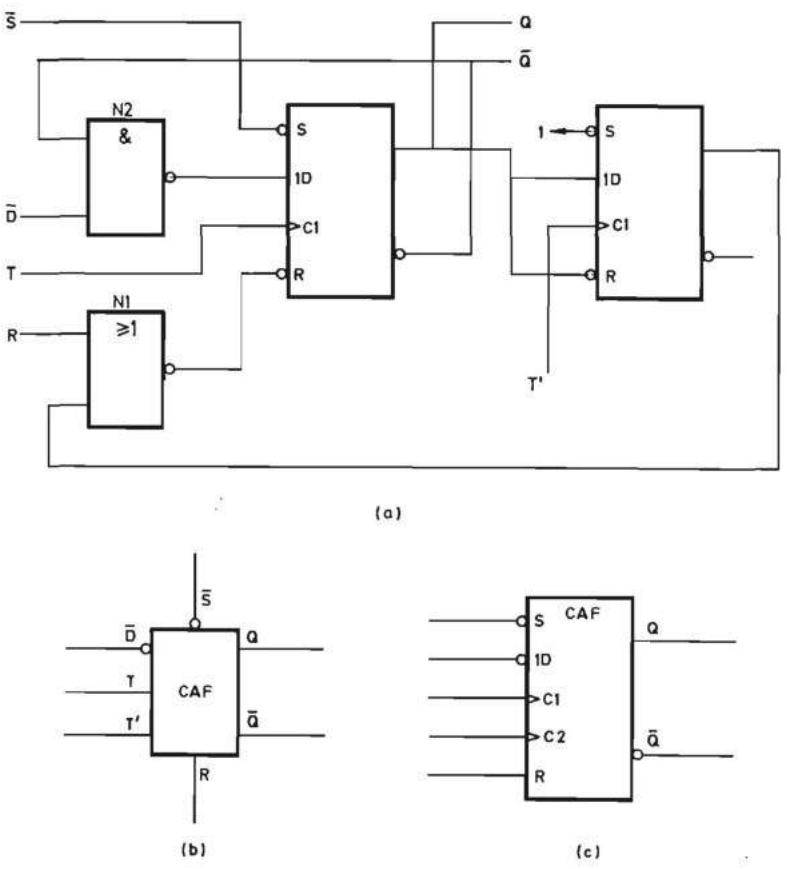


FIGURA 6.141.—**Célula CAF realizada con biestables *D* activados por flancos: *a*) Esquema del circuito; *b*) Símbolo lógico no normalizado; *c*) Símbolo lógico normalizado.**

se encuentra en estado cero, la salida Q_2 adopta también este estado debido a que su entrada de puesta a cero R_2 está activada. Al subir Q_1 al estado uno desaparece la puesta a cero del biestable Q_2. Este pasa al estado uno cuando se aplica un flanco activo a T' y a través de N_1 produce una puesta a cero de Q_1 que, a su vez, vuelve a realizar una puesta a cero de Q_2. Las entradas R y S permiten realizar una puesta a cero y una puesta a uno respectivamente de la célula.

La entrada de puesta a uno ($\overline{S}$) es activa con un cero lógico y la de puesta a cero (R) es activa con un uno lógico.

En las figuras 6.140*b* y 6.140*c* se representan los símbolos lógicos no normalizado y normalizado respectivamente, correspondientes a la célual CAF de la figura 6.140*a*.

En la figura 6.141*a* se representa el esquema de la célula CAF realizada con biestables D activados por flancos. La principal diferencia con respecto a la figura 6.140*a* es la necesidad de la puerta $N2$ que realimenta la salida Q_1 para que, una vez que Q pasa al estado uno, sólo vuelva al estado cero al aplicar un flanco activo a T'. La entrada D de la célula resulta invertida ($\overline{D}$) porque la célula pasa al estado uno cuando T pasa de cero a uno y la entrada D se encuentra en nivel cero. El símbolo lógico de la célula CAF obtenida se representa en las figuras 6.141*b* y 6.141*c*. Este símbolo coincide con el de las figuras 6.140*b* y 6.140*c* excepto en que la entrada D ha sido sustituida por $\overline{D}$.

6.3.4 Representación gráfica de los biestables síncronos

En los apartados anteriores se analizaron los símbolos lógicos de los diferentes biestables síncronos. En este apartado se vuelve a insistir sobre ello para indicar la especial trascendencia de los símbolos normalizados que permiten mediante su observación deducir la forma de operación de los biestables sin necesidad de ninguna información adicional.

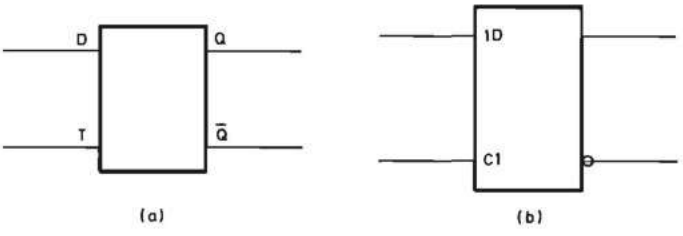


FIGURA 6.142.—**Símbolos lógicos del biestable *D* cerrojo sincronizado mediante el nivel uno de la entrada *T* (*C*1): *a*) No normalizado; *b*) Normalizado.**

Como resumen, en las figuras 6.142 a 6.152 se representan los símbolos no normalizados y normalizados correspondientes a diferentes tipos de biestables síncronos y células CAF. El lector puede comprobar las ventajas que los últimos presentan sobre los primeros.

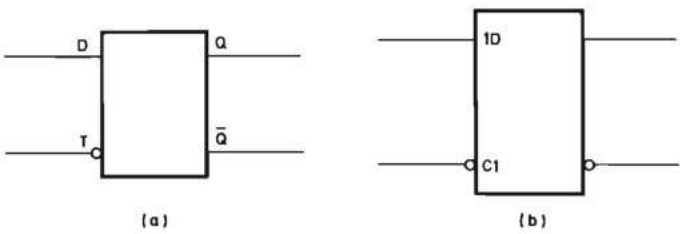


FIGURA 6.143.—**Símbolos lógicos del biestable *D* cerrojo sincronizado mediante el nivel cero de la entrada *T* (*C*1): *a*) No normalizado; *b*) Normalizado.**

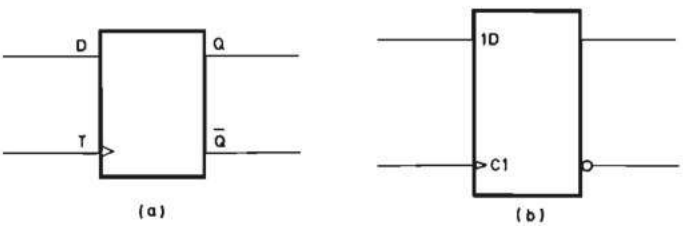


FIGURA 6.144.— **Símbolos lógicos del biestable *D* activado por flancos de subida: *a*) No normalizado; *b*) Normalizado.**

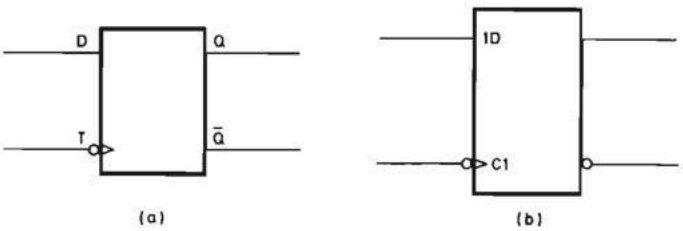


FIGURA 6.145.— **Símbolos lógicos del biestable *D* activado por flancos de bajada: *a*) No normalizado; *b*) Normalizado.**

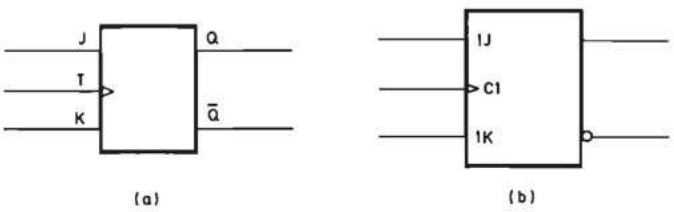


FIGURA 6.146.— **Símbolos lógicos del biestable *J-K* activado por flancos de subida: *a*) No normalizado; *b*) Normalizado.**

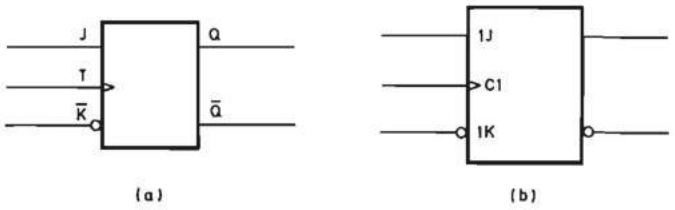


FIGURA 6.147.— **Símbolos lógicos del biestable *J-K̄* activado por flancos de subida: *a*) No normalizado; *b*) Normalizado.**

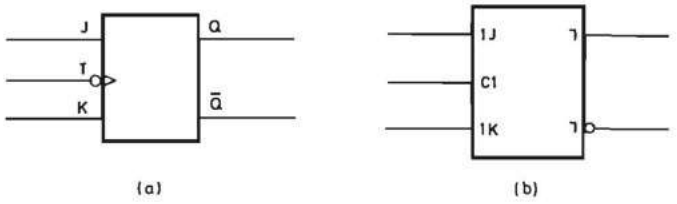


FIGURA 6.148.— **Símbolos lógicos del biestable *J-K* principal-subordinado activado por impulsos positivos: *a*) No normalizado; *b*) Normalizado.**

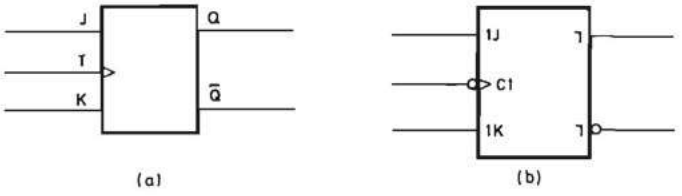


FIGURA 6.149.— **Símbolos lógicos del biestable *J-K* principal-subordinado con enclavamiento, activado por impulsos negativos: *a*) No normalizado; *b*) Normalizado.**

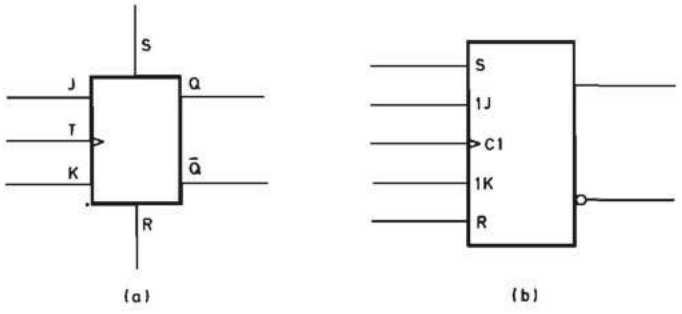


FIGURA 6.150.— **Símbolos lógicos del biestable *J-K* activado por flancos de subida con puesta a cero y puesta a uno asíncronas activas con nivel uno: *a*) No normalizado; *b*) Normalizado.**

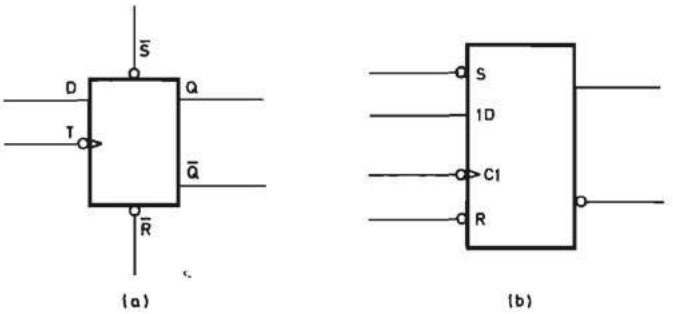


FIGURA 6.151.— **Símbolos lógicos del biestable *D* activado por flancos de bajada con puesta a cero y puesta a uno activas con nivel cero: *a*) No normalizado; *b*) Normalizado.**

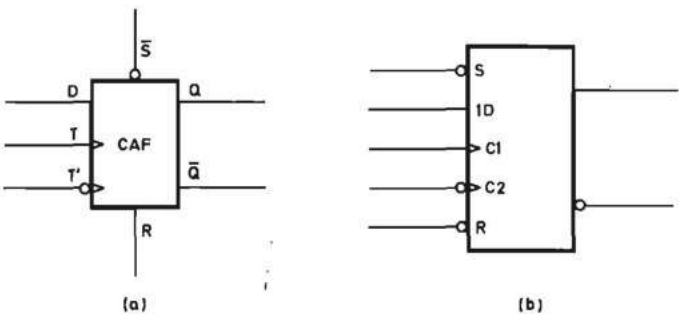


FIGURA 6.152.—**Símbolos lógicos de la célula CAF con entrada *T*(*C*1) activa con flancos de subida y *T'*(*C*2) activa con flancos de bajada.**

6.3.5 Sistemas secuenciales síncronos de control

6.3.5.1 Introducción. En el apartado 6.2.2.3 se estudiaron los sistemas secuenciales asíncronos y su realización mediante biestables *S-R* activados por niveles. En la figura 6.153 se representa el diagrama general de bloques de un sistema secuencial asíncrono realizado con biestables *S-R* correspondiente a un autómata de Moore. Este sistema evoluciona de una forma autónoma. Dado un estado de entrada (x_1, ..., x_n) y un estado interno (Q_1, ..., Q_m) estable para dicho estado de entrada, los valores de las variables S_i y R_i (para i de 1 a n) toman un valor tal que los biestables *SR* permanecen en un estado fijo.

Al cambiar una variable de entrada, el sistema ha de ir a un nuevo estado interno que puede ser distinto del actual. En este caso el sistema combinacional hace que para el nuevo estado de entrada y el estado interno actual cambien las S_i y R_i adecuadas para que se produzca el nuevo estado interno. Si este nuevo estado interno es estable, las entradas S_i y R_i permanecen invariables hasta que se produzca un nuevo cambio en el vector de entrada. Si el nuevo estado interno no es estable, se produce un nuevo cambio en la entradas S_i y R_i y así sucesivamente hasta alcanzar un estado interno estable.

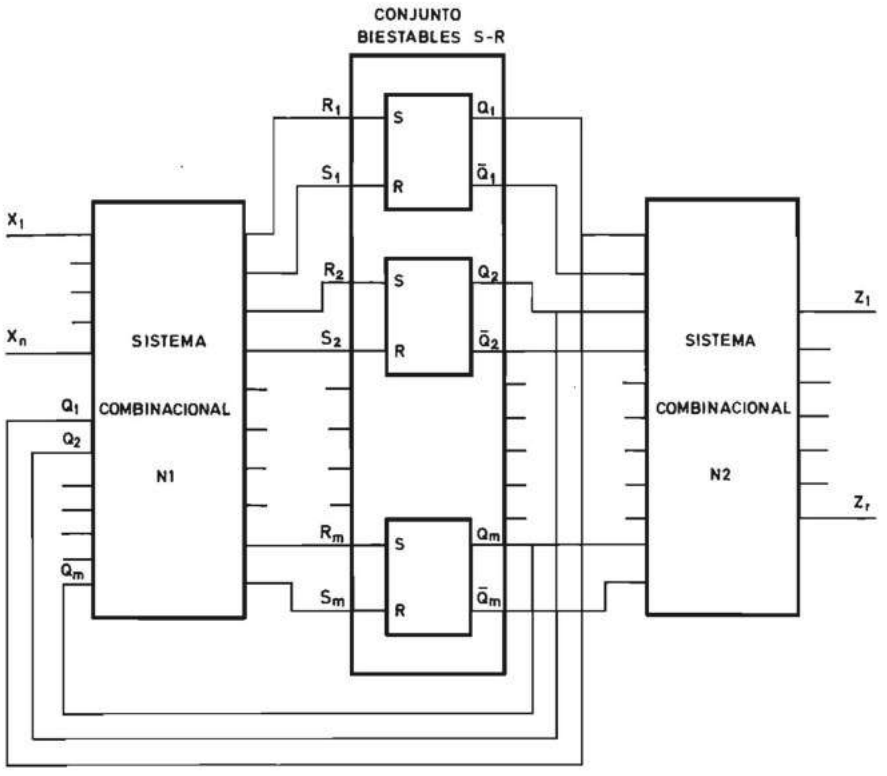


FIGURA 6.153.—**Diagrama de bloques de un sistema secuencial asíncrono realizado con biestables *R-S*.**

El sistema asíncrono va, por lo tanto, evolucionando entre estados internos de una forma autónoma cuando se produce un cambio de las entradas. Su forma de operación se puede representar mediante un grafo denominado diagrama de flujo como el indicado en la figura 6.154.

En ella se representa un autómata asíncrono con cuatro estados estables E_1 a E_4. Las evoluciones entre estados se realizan al aparecer determinadas variables de entrada. Por ejemplo, si el autómata está en el estado E_2 y aparece la combinación de entrada X_j, pasa al estado E_3. Un ejemplo práctico de un sistema de este tipo es el representado mediante el diagrama de la figura 6.12.

La asimilación de los conceptos expuestos en los párrafos anteriores, que fueron desarrollados en el apartado 6.2, permitirá al lector comprender el funcionamiento de un sistema secuencial síncrono cuyo diagrama de bloques se representa en la figura 6.155 en sus dos versiones: no normalizada (*a*) y normalizada (*b*). Dicho sistema está constituido fundamentalmente por una unidad formada por biestables síncronos activados por flancos de cualquiera de los tipos descritos en el apartado 6.3.2 (en la figura 6.155 se han utilizado los del tipo *J-K* activados por flancos) y un sistema combinacional *N*1 cuyas entradas son las salidas de los biestables Q_1 a Q_m y las entradas exteriores x_1 a x_n, y cuyas salidas se conectan a las

entradas de información de los biestables (en la figura las entradas J y K por ser los biestables del tipo J-K activados por flancos). Las entradas T ($C1$) de todos los biestables se conectan entre sí y a la salida del generador de impulsos. En el diagrama de bloques de la figura 6.155*b* se representa el conjunto de m biestables síncronos mediante el símbolo de un registro formado por m biestables J-K con una entrada común del generador de impulsos $C1$.

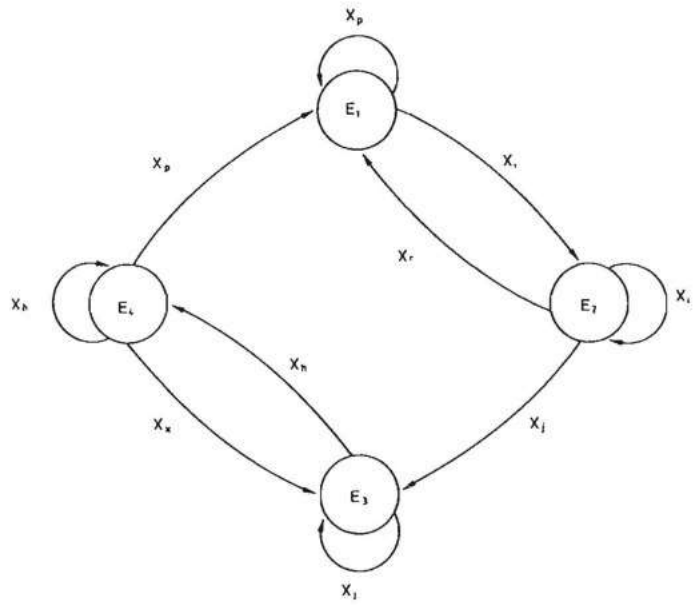
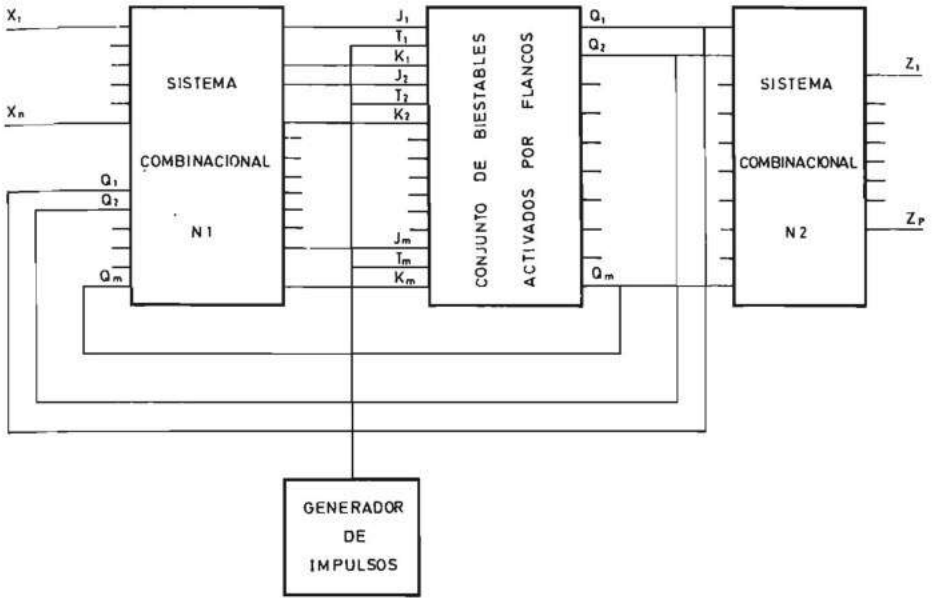


FIGURA 6.154.—**Diagrama de flujo de un sistema secuencial asíncrono caracterizado por niveles.**

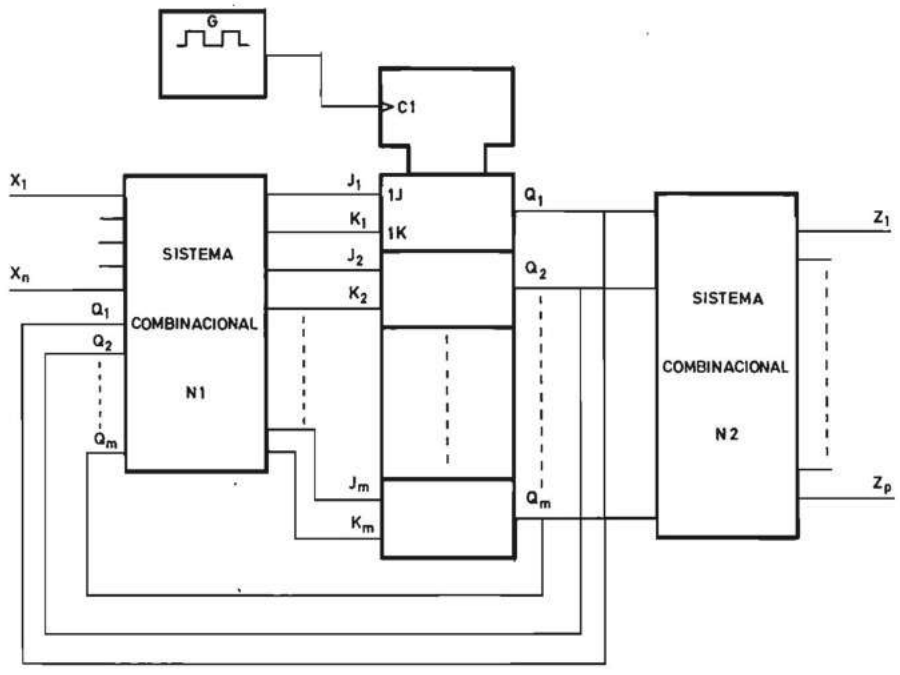
La forma de operación de este sistema es análoga a la del asíncrono de la figura 6.153 con la diferencia fundamental de que los cambios de estado interno (cambios de estado de los biestables Q_1 a Q_m) se producen cuando se aplica un flanco activo (que puede ser el de subida o el de bajada) a la entrada T de los biestables. Aunque las entradas x_1 a x_n cambien en un instante determinado, dicho cambio no afecta al estado interno hasta que se aplica el impulso del generador a las entradas T. Los cambios de estado interno están sincronizados con los impulsos del generador, por lo cual el autómata recibe el adjetivo de síncrono.

El funcionamiento de un autómata secuencial síncrono se puede representar mediante un diagrama de flujo similar al de la figura 6.154. La única diferencia de uno con respecto a otro es que en el sistema síncrono los cambios de estado se producen al aparecer un flanco activo de los impulsos del generador, mientras que en los sistemas asíncronos el cambio de las entradas produce de inmediato el cambio de estado interno.

Un ejemplo permitirá aclarar mejor los conceptos que acabamos de exponer:



(a)



(b)

FIGURA 6.155.— **Diagramas de bloques de un sistema secuencial síncrono: *a)* No normalizado; *b)* Normalizado.**

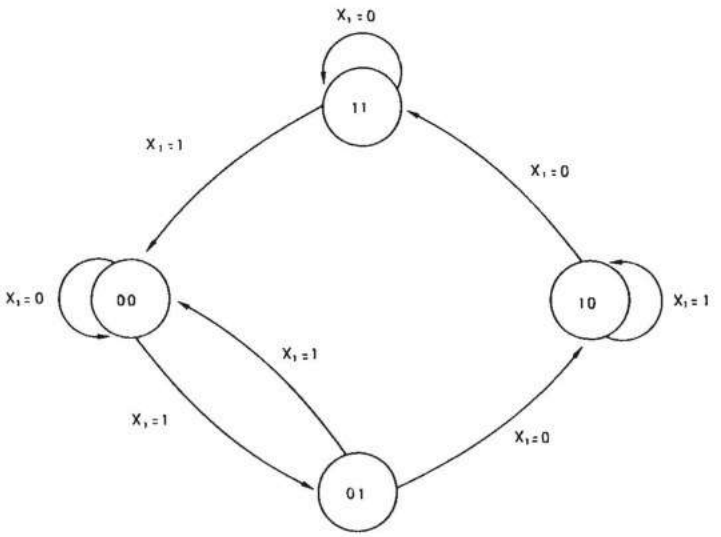


FIGURA 6.156.—**Ejemplo de diagrama de flujo de un sistema secuencial síncrono.**

Ejemplo 6.4:

Sea un sistema secuencial síncrono con una variable de entrada x_1 y que ha de realizar el diagrama de flujo de la figura 6.156. Las variables de estado interno se denominan Q_0 y Q_1.

En el diagrama de la figura 6.156 se observa que, a partir de un estado interno determinado, el sistema cambia de estado al aplicar impulsos de sincronismo en función del estado de las variables de entrada. Por ejemplo, si el sistema se encuentra en el estado interno 00, al aplicar un impulso de sincronismo permanece en él si la variable de entrada x_1 es 0 y pasa al 01 si la variable de entrada x_1 es 1.

El diagrama de la figura 6.156 se puede representar mediante la tabla de funcionamiento indicada en la tabla 6.19, en la que las variables independientes son las variables de estado interno en el instante t (antes de aplicar el impulso de sincronismo) y la variable de entrada x_1, y las variables dependientes son las variables de estado interno después de aplicar el citado impulso, es decir, en el instante $t+1$.

La tabla 6.19 se deduce fácilmente a partir del diagrama de la figura 6.156. Para cada estado interno en el instante t y cada combinación de las variables de entrada se observa en la figura 6.156 cuál es el nuevo estado interno en el instante $t+1$. El estado interno y la combinación de las variables de entrada forman una de las combinaciones de la izquierda de la tabla 6.19 y el nuevo estado interno la combinación de la misma línea de la derecha. Por ejemplo, si el estado interno es 10 y la variable x_1 es cero, el nuevo estado interno es 11. Esto se indica en la quinta línea de la tabla 6.19.

A partir de la tabla 6.19 es necesario deducir el nivel lógico que deben tener las variables J y K para que se produzcan las transiciones adecuadas entre estados. Para ello en la tabla 6.20*a* se representan los niveles lógicos que deben adoptar las entradas J_i y K_i para cada una de las combinaciones de Q_{it} y Q_{it+1}. El lector no debe tener

Q_{1t}	Q_{0t}	X_1	Q_{1t+1}	Q_{0t+1}
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

TABLA 6.19.—**Tabla de funcionamiento de un sistema secuencial síncrono que realiza el diagrama de flujo de la figura 6.156.**

Q_{it}	Q_{it+1}	J_i	K_i
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(a)

Q_1	Q_0	X_1	J_1	K_1	J_0	K_0
0	0	0	0	X	0	X
0	0	1	0	X	1	X
0	1	0	1	X	X	1
0	1	1	0	X	X	1
1	0	0	X	0	1	X
1	0	1	X	0	0	X
1	1	0	X	0	X	0
1	1	1	X	1	X	1

(b)

TABLA 6.20

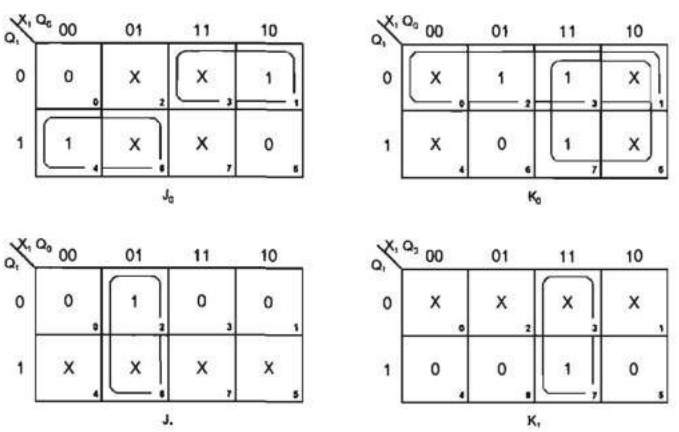


FIGURA 6.157

ninguna dificultad para deducir dicha tabla a partir de la descripción del comportamiento del biestable *J-K* principal-subordinado descrito en el apartado 6.3.2.2.1.1. A partir de las tablas 6.19 y 6.20*a* se obtiene la tabla 6.20*b* y a partir de esta última las tablas de Karnaugh de la figura 6.157 de la cual se deducen las ecuaciones:

$$\begin{aligned} J_0 &= x_1 \bar{Q}_1 + \bar{x}_1 Q_1 \\ K_0 &= x_1 + \bar{Q}_1 \\ J_1 &= \bar{x}_1 Q_0 \\ K_1 &= x_1 Q_0 \end{aligned}$$

Realizando estas ecuaciones con puertas NO-O resulta:

$$\begin{aligned} J_0 &= \overline{\overline{\bar{x}_1 + Q_1} + \overline{x_1 + \bar{Q}_1}} \\ K_0 &= \overline{\overline{x_1 + \bar{Q}_1}} \\ J_1 &= \overline{x_1 + \bar{Q}_0} \\ K_1 &= \overline{\bar{x}_1 + \bar{Q}_0} \end{aligned}$$

En la figura 6.158 se representa el esquema del sistema secuencial síncrono obtenido cuya forma de funcionamiento es la indicada en el diagrama de flujo de la figura 6.156. Una de las principales ventajas de los sistemas secuenciales síncronos sobre los asíncronos es la ausencia de transiciones críticas. En efecto, aunque al producirse el paso de un estado interno a otro cambien más de una variable de estado interno, no importan los estados intermedios por los que se pase si la separación entre impulsos de sincronismo es tal que cuando se aplica un nuevo impulso ya han cambiado todas las variables. Esta propiedad simplifica el problema de la codificación de estados internos que trataremos en el apartado 6.3.5.5.3.

Las variables de entrada del sistema secuencial síncrono han de estar sincronizadas con los impulsos del generador de impulsos o reloj. En general, cambiarán después el flanco activo que introduce la información en los biestables, y por ello estos últimos pueden ser de activación por flancos o principal-subordinado y ser activados con los flancos de subida o bajada. Para distinguirlos de los biestables síncronos activados por niveles se coloca el indicativo de flanco en la entrada *C*.

El sistema secuencial síncrono de la figura 6.155 es un autómata síncrono general cuyo diagrama de flujo puede ser cualquiera y depende de la tabla de verdad del circuito combinacional.

Pero existe un conjunto de casos particulares que merece un estudio especial por dos razones:

a) Poseer un conjunto de aplicaciones que han motivado su realización en circuito integrado de mediana o gran escala de integración.

b) Poder ser utilizados como bloques funcionales para la síntesis de cualquier sistema secuencial síncrono.

En el apartado siguiente se analizan dichos sistemas.

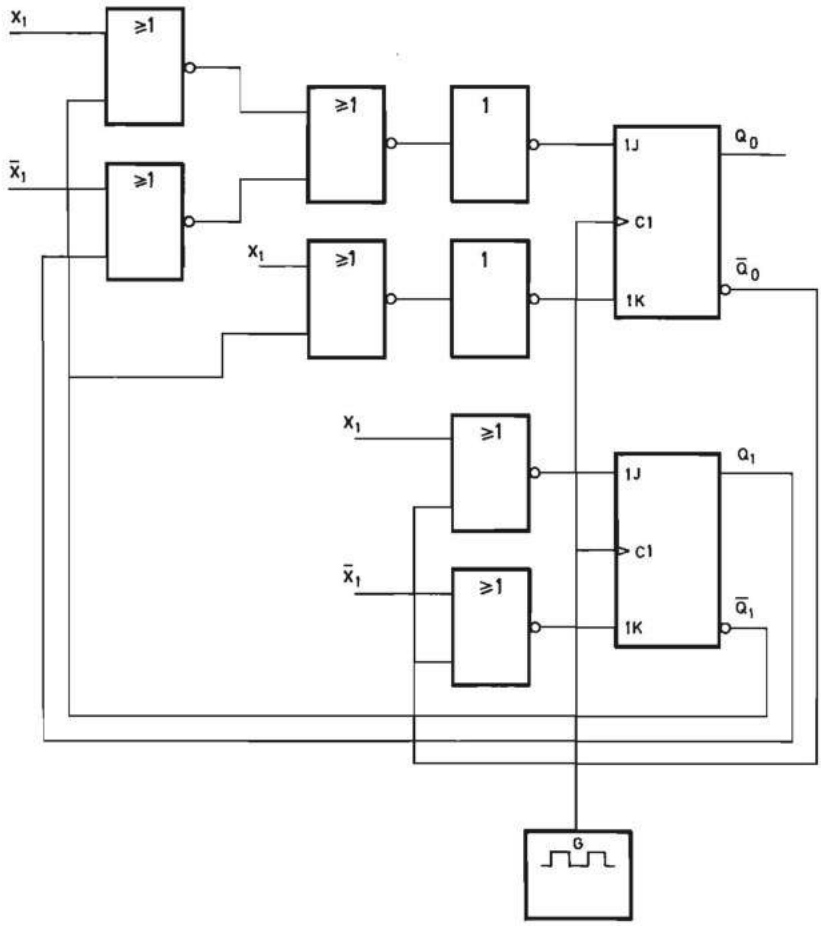
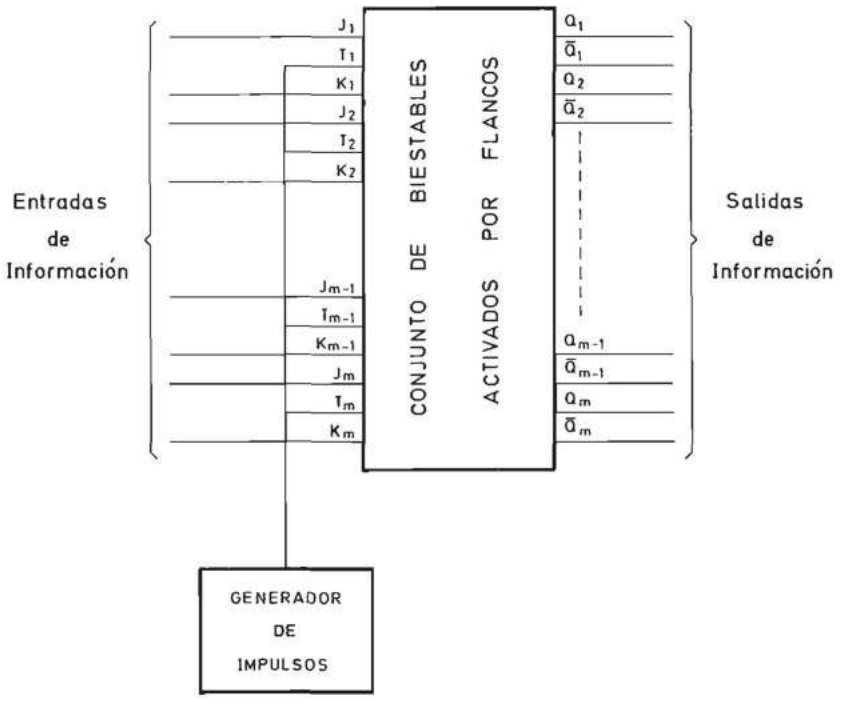


FIGURA 6.158.—**Esquema del sistema secuencial síncrono realizado con biestables *J-K* activados por flancos de subida que realiza el diagrama de flujo de la figura 6.156.**

6.3.5.2 Sistemas secuenciales síncronos de control de aplicación general

6.3.5.2.1 Registros de entrada y salida en paralelo. Si en el sistema de la figura 6.155 se suprime el sistema combinacional *N*1 y las salidas de cada biestable se disponen accesibles desde el exterior, al igual que las entradas, se obtiene un registro de información de entrada y salida en paralelo representado en la figura 6.159. Por tratarse de un sistema de aplicación general existe como bloque funcional en escala de integración media en todas las tecnologías. Por ser su finalidad el almacenamiento de información se suele realizar con biestables *D* activados por flancos, los cuales presentan la ventaja sobre los *J-K* de poseer sólo una entrada. El diagrama



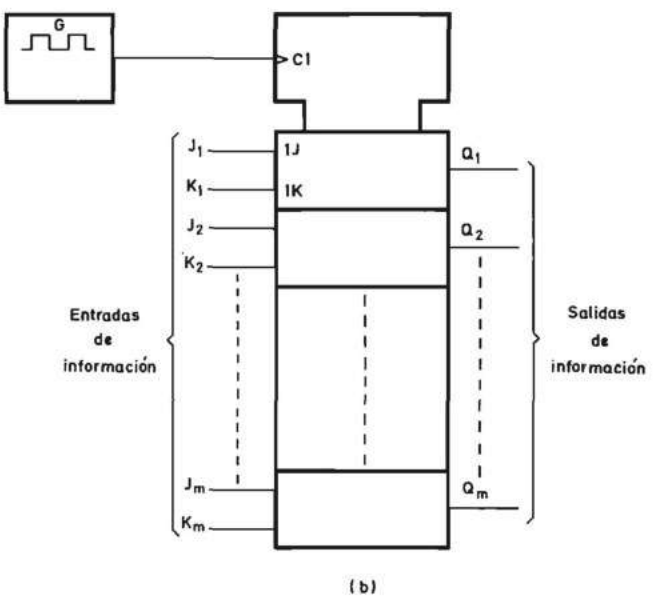
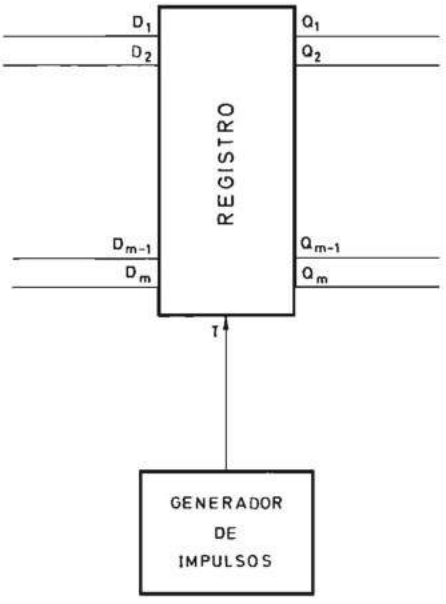
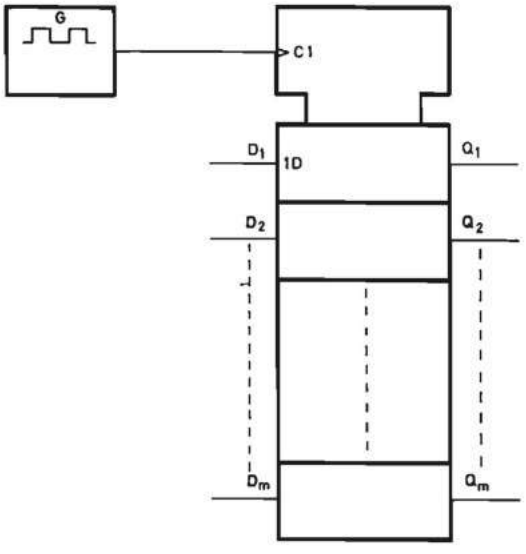


FIGURA 6.159.—**Diagrama de bloques de un registro síncrono de entrada y salida paralelo realizado con biestables *J-K*: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**



(a)



(b)

FIGURA 6.160.— **Diagrama de bloques de un registro de entrada y salida en paralelo realizado con biestables *D*: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**

de bloques correspondiente se representa en la figura 6.160. Para reducir la complejidad de los esquemas, la entrada y la salida de información de estos registros se suelen representar mediante un solo hilo cruzado por una línea inclinada sobre la que se indica el número *m* de biestables que lo constituyen (figura 6.161).

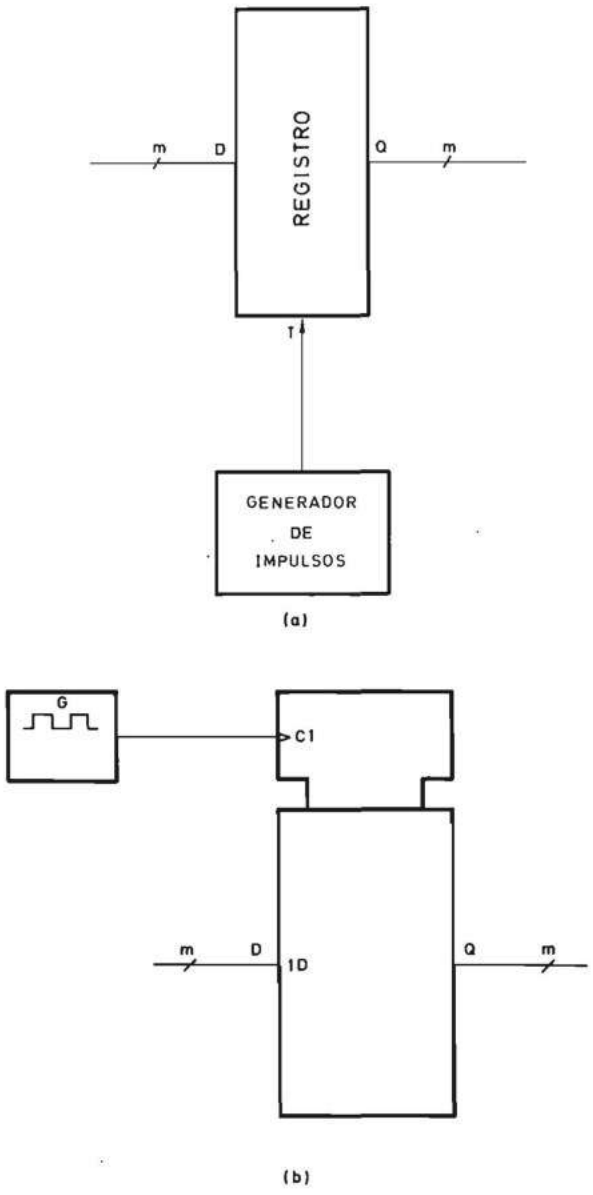


FIGURA 6.161.—**Diagrama de bloques de un registro de entrada y salida en paralelo realizado con biestables *D*: *a*) Símbolo lógico no normalizado. *b*) Símbolo lógico normalizado.**

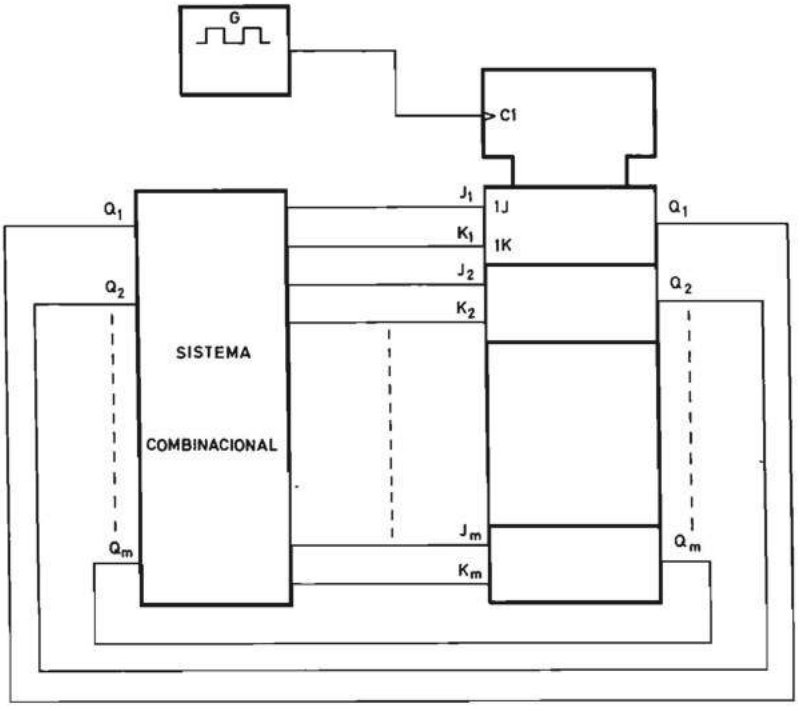


FIGURA 6.162.—**Diagrama de bloques de un contador síncrono.**

6.3.5.2.2 Contadores síncronos. Supóngase que el sistema combinacional *N*1 de la figura 6.155 no posee las entradas exteriores x_1 a x_n. El sistema resultante se representa en la figura 6.162. El diagrama de flujo de un sistema de este tipo que posea m biestables tiene $2^m = n$ estados internos y se representa en la figura 6.163. Este sistema cambia de estado cada vez que se aplica a la entrada T de sus biestables un impulso de sincronismo. El estado en que se encuentra en cada instante depende del número de impulsos de sincronismos que se ha aplicado hasta ese momento. Por lo tanto, este sistema secuencial síncrono constituye un contador.

La figura 6.162 se puede representar también de forma más simplificada indicando las m salidas de los biestables y las m entradas mediante un segmento sobre el que se coloca el indicativo m. Se obtiene así la figura 6.164.

Por ser los contadores sistemas secuenciales síncronos de aplicación general, se estudiarán con mayor detenimiento en el apartado 6.3.5.4.3 posteriormente. Pero antes se analizarán diversas variantes que se pueden realizar en la figura 6.162. Al sistema representado en dicha figura se le puede añadir una variable de entrada I, que cuando adopte un determinado nivel lógico coloque las variables de entrada de los biestables en un estado tal que los impulsos aplicados a éstos no hagan cambiar su estado (si los biestables son *J-K*, la variable I hará que $J_i = K_i = 0$ para todos los biestables). Se obtiene así un sistema cuyo esquema es el de la figu-

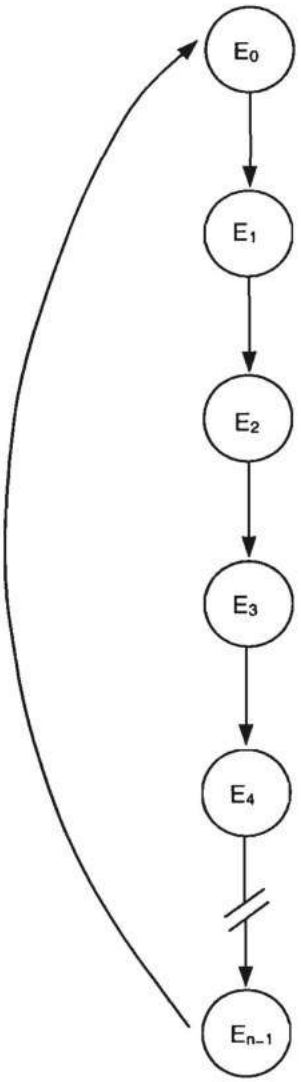


FIGURA 6.163.—**Diagrama de flujo de un contador síncrono.**

ra 6.165 y su diagrama de flujo el representado en la figura 6.166. El sistema obtenido constituye un contador con entrada de inhibición.

Al sistema combinacional de la figura 6.162 se le puede añadir una entrada *A/D* (fig. 6.167) que, según su estado sea el uno o el cero lógico, haga que a partir de un estado determinado el sistema vaya al siguiente o al anterior. Se obtiene así un contador síncrono cuyo diagrama de flujo es el representado en la figura 6.168, y que recibe el nombre de reversible porque realiza el contaje en sentido ascendente o descendente según el valor de la variable *A/D*.

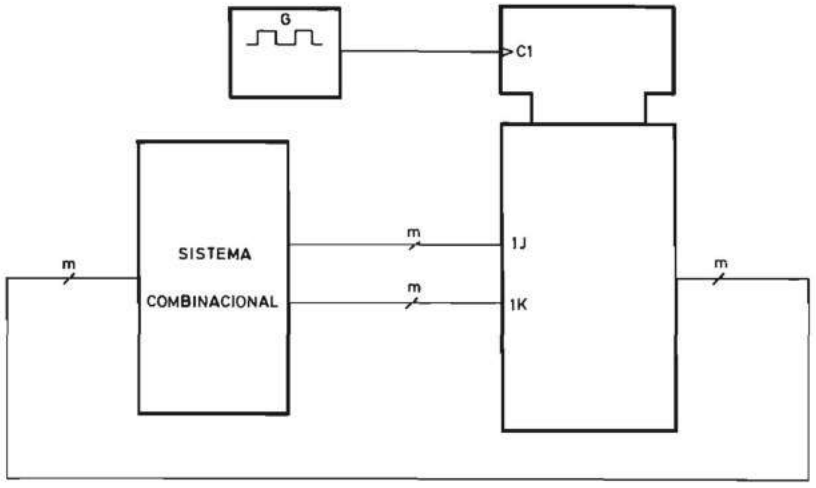


FIGURA 6.164.—**Diagrama de bloques simplificado de un contador síncrono realizado con biestables *J-K*.**

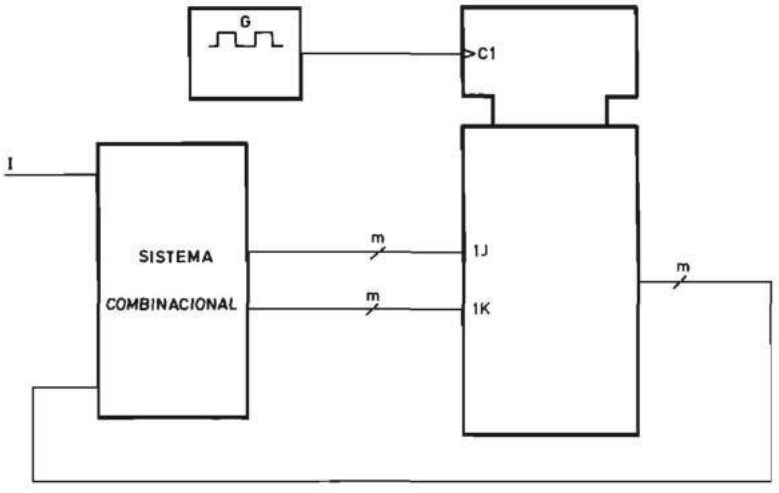


FIGURA 6.165.—**Diagrama de bloques de un contador síncrono realizado con biestables *J-K* que posee una entrada de inhibición.**

Se puede realizar un contador que posea ambas variables de entrada *I* y *A/D*, es decir, que sea reversible y al mismo tiempo se pueda inhibir su evolución. Su diagrama de flujo será el de la figura 6.169 y su esquema el de la figura 6.170.

También resulta de gran interés, tal como se verá en apartados posteriores, dotar a los contadores síncronos de la posibilidad de que en ellos se pueda introducir información exterior en paralelo al aplicar el impulso de sincronismo en las entradas *T*. Esta opción se puede añadir a cualquiera de los esquemas de las figuras

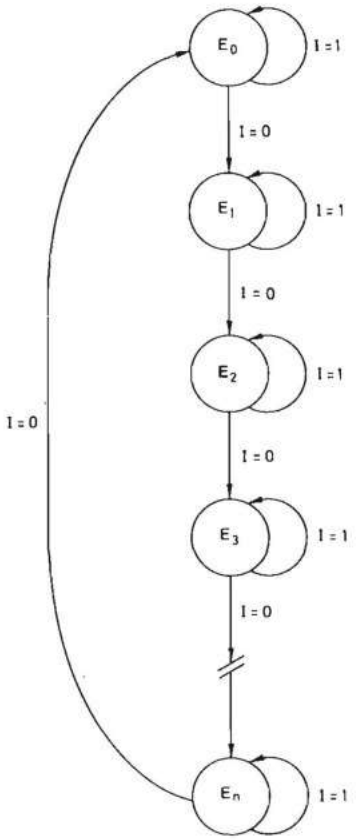


FIGURA 6.166.—**Diagrama de flujo de un contador síncrono con entrada de inhibición.**

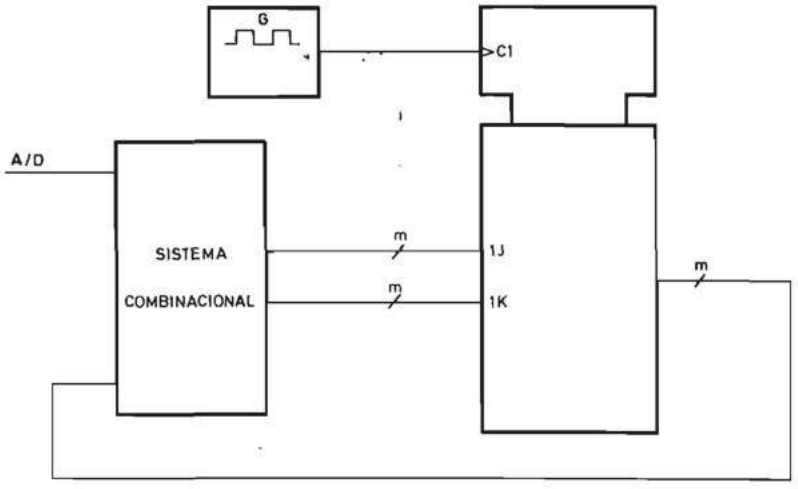


FIGURA 6.167.—**Diagrama de bloques de un contador síncrono reversible realizado con biestables *J-K*.**

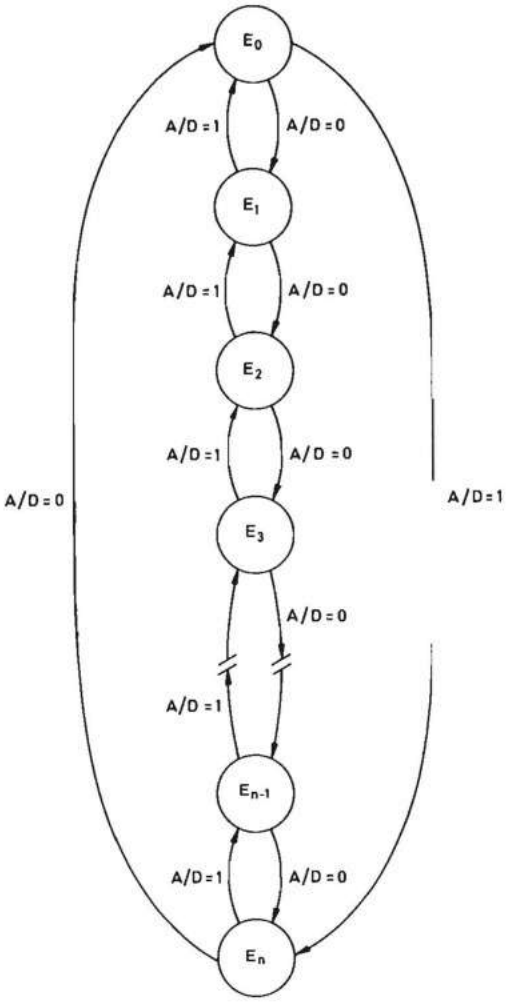


FIGURA 6.168.—**Diagrama de flujo de un contador síncrono reversible.**

6.164, 6.165, 6.167 y 6.170. En la figura 6.171 se representa el esquema de la figura 6.164 modificado para dotarlo de entradas en paralelo. Entre el sistema combinacional y las entradas de los biestables se colocan m multiplexores de dos canales con una variable de selección común C/P. Uno de los canales de los multiplexores se conecta a las salidas del sistema combinacional y el otro al exterior. Según el estado de la variable C/P, los impulsos aplicados en T harán que el contador pase al estado siguiente o que en él se introduzca la información externa en paralelo.

En los distintos esquemas de contadores que se han analizado se puede utilizar cualquiera de los biestables síncronos activados por flancos estudiados en el apar-

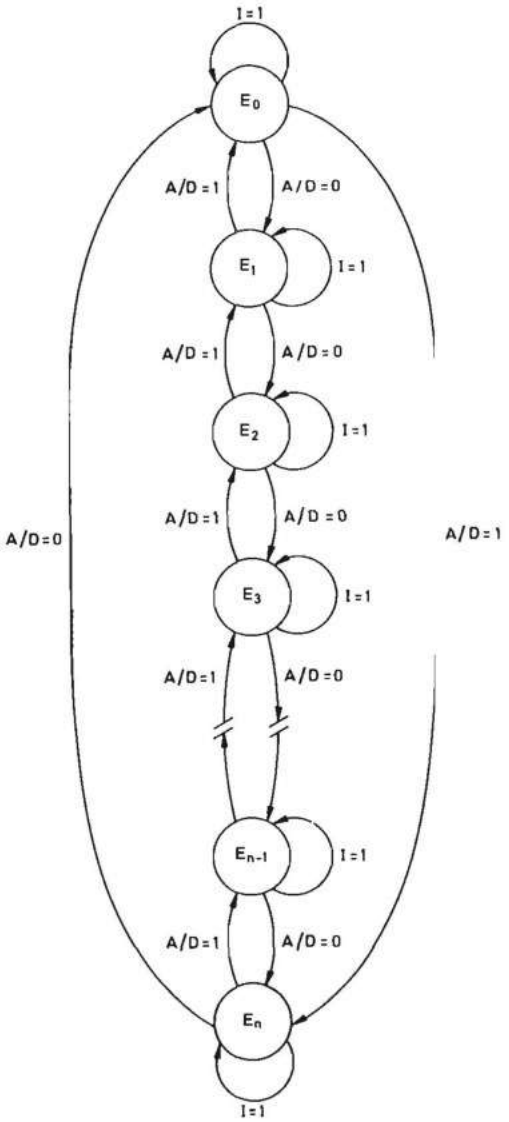


FIGURA 6.169.—**Diagrama de flujo de un contador síncrono reversible con entrada de inhibición.**

tado 6.3.2.2. Como se estudió en dicho apartado, estos biestables suelen poseer entradas de puesta a cero y a uno asíncronas. Por ello a los contadores se les puede añadir una o más variables de puesta en estado inicial que se obtienen conectando entre sí todas las variables de puesta a cero o puesta a uno de los distintos biestables. En la figura 6.172 se representa el diagrama de bloques de un contador con entrada de puesta en estado inicial asíncrona.

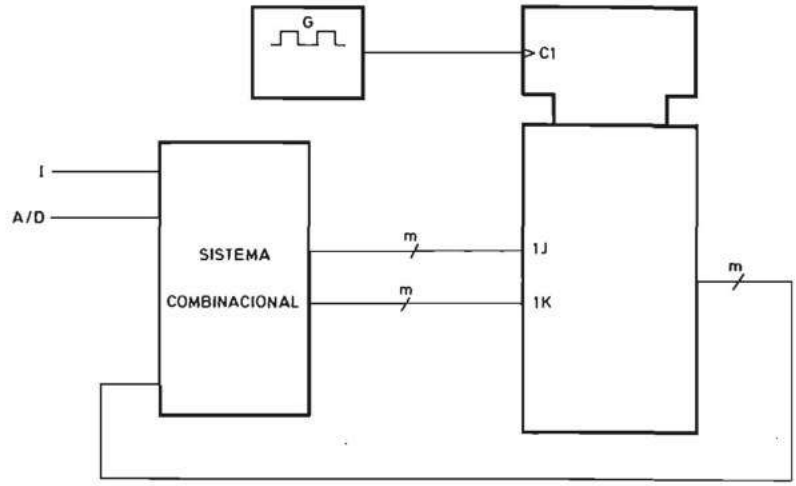


FIGURA 6.170.—**Diagrama de bloques de un contador síncrono reversible con entrada de inhibición realizado con biestables *J-K*.**

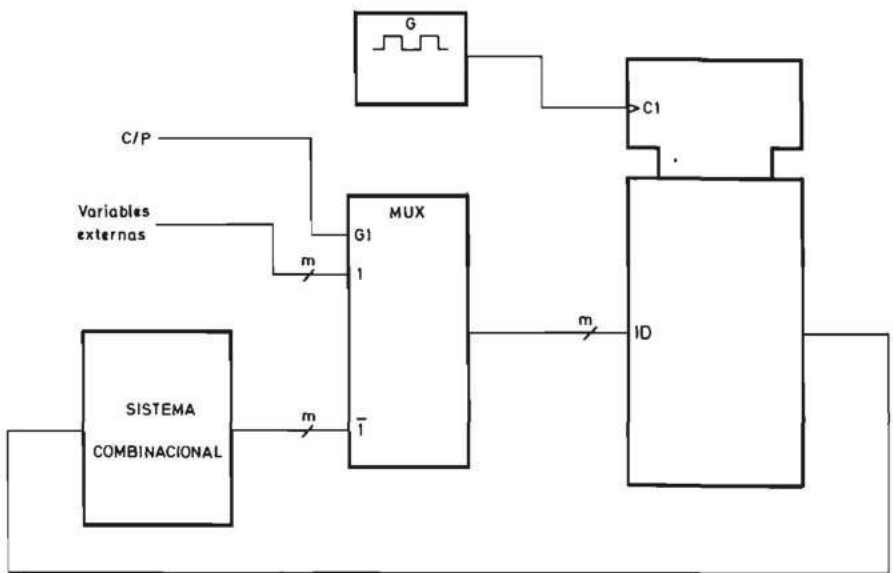


FIGURA 6.171.—**Diagrama de bloques de un contador síncrono con entradas en paralelo síncronas realizado con biestables *D*.**

6.3.5.2.3 Registros de desplazamiento. Si en el sistema de la figura 6.155 se suprime el sistema combinacional $N1$ y las salidas de cada biestable Q_i se conectan directamente a las entradas del biestable Q_{i+1}, se obtiene el sistema de la figura 6.173*a*. Si se disponen al exterior las entradas del primer biestable y las sali-

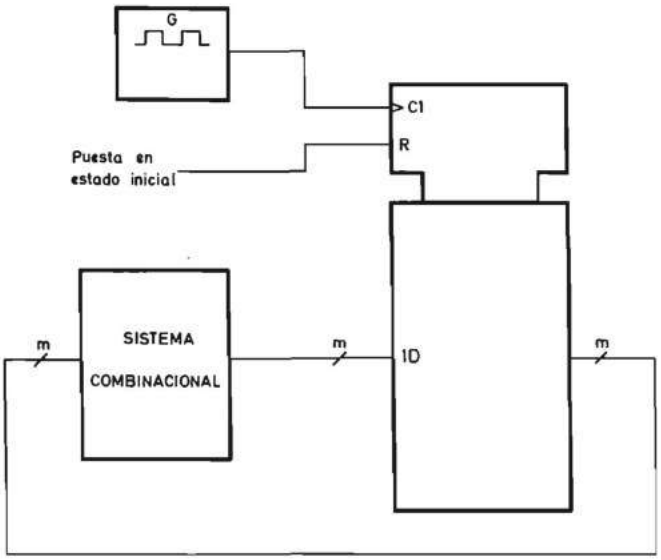


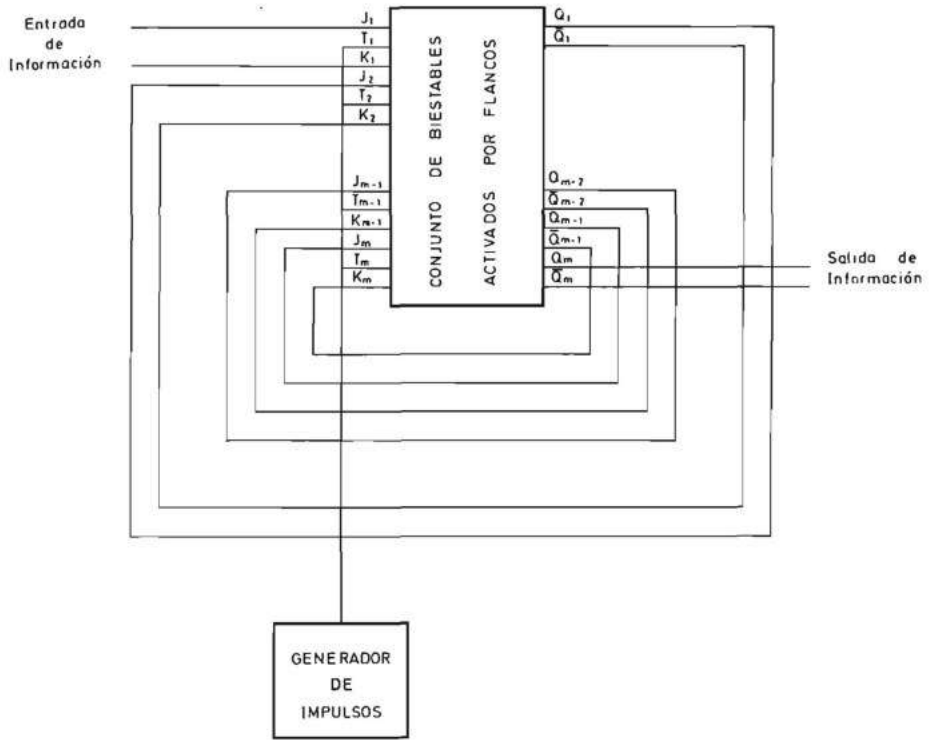
FIGURA 6.172.—**Diagrama de bloques de un contador con puesta en estado inicial asíncrona, realizado con biestables *D*.**

das del último, este sistema es un registro de desplazamiento: la información contenida en cada biestable se desplaza al siguiente al aplicar un impulso de sincronismo. Por tratarse de un sistema de aplicación general su estudio se realiza también en el apartado 6.3.5.4.4. En la figura 6.173*a* se utilizan biestables *J-K* y por ello las salidas Q y $\overline{Q}$ de cada biestable se conectan respectivamente a las entradas J y K del siguiente. El conexionado se simplifica utilizando biestables tipo D que resultan más adecuados para realizar registros de desplazamiento. El esquema correspondiente se representa en la figura 6.174*a*.

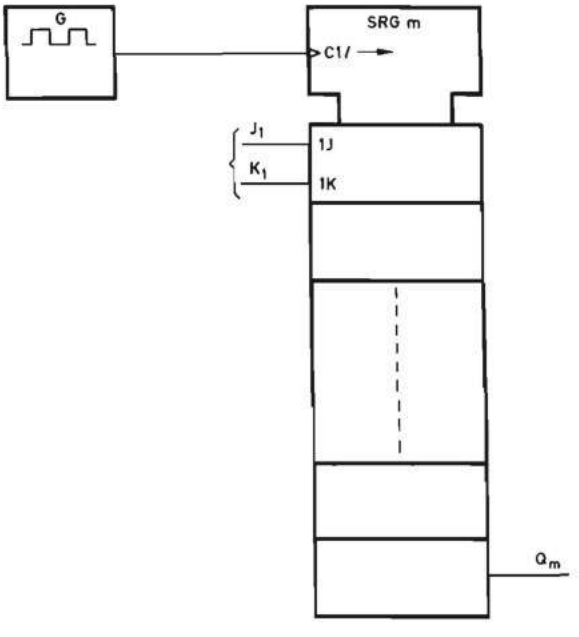
Los diagramas de las figuras 6.173*a* y 6.174*a* resultan complejos en lo que se refiere a su realización. Por ello se pueden sustituir por los símbolos normalizados representados en las figuras 6.173*b* y 6.174*b*, que se caracterizan por poseer el indicativo SRGm que indica que se trata de un registro de desplazamiento (shift register) de m bits con una entrada de impulsos de desplazamiento, que hacen que se desplace la información hacia la derecha ($C1/\rightarrow$), y una de información ($1J$-$1K$ en la figura 6.173*b* y $1D$ en la 6.174*b*).

Al igual que los contadores, los registros de desplazamiento pueden también poseer entradas de inhibición, ser reversibles o tener la posibilidad de introducir en ellos información en paralelo. El lector puede, a partir de lo estudiado, deducir los esquemas correspondientes.

6.3.5.3 Sistemas secuenciales síncronos asincronizados. En el apartado 6.3.5.1 se analizaron los sistemas secuenciales síncronos cuyo diagrama de bloques se representa en la figura 6.155 y que se caracterizan porque los impulsos del generador se



(a)



(b)

FIGURA 6.173.—**Diagrama de bloques de un registro de desplazamiento realizado con biestables *J-K*: *a*) Con indicación de todas las conexiones entre los biestables; *b*) Mediante un símbolo lógico normalizado.**

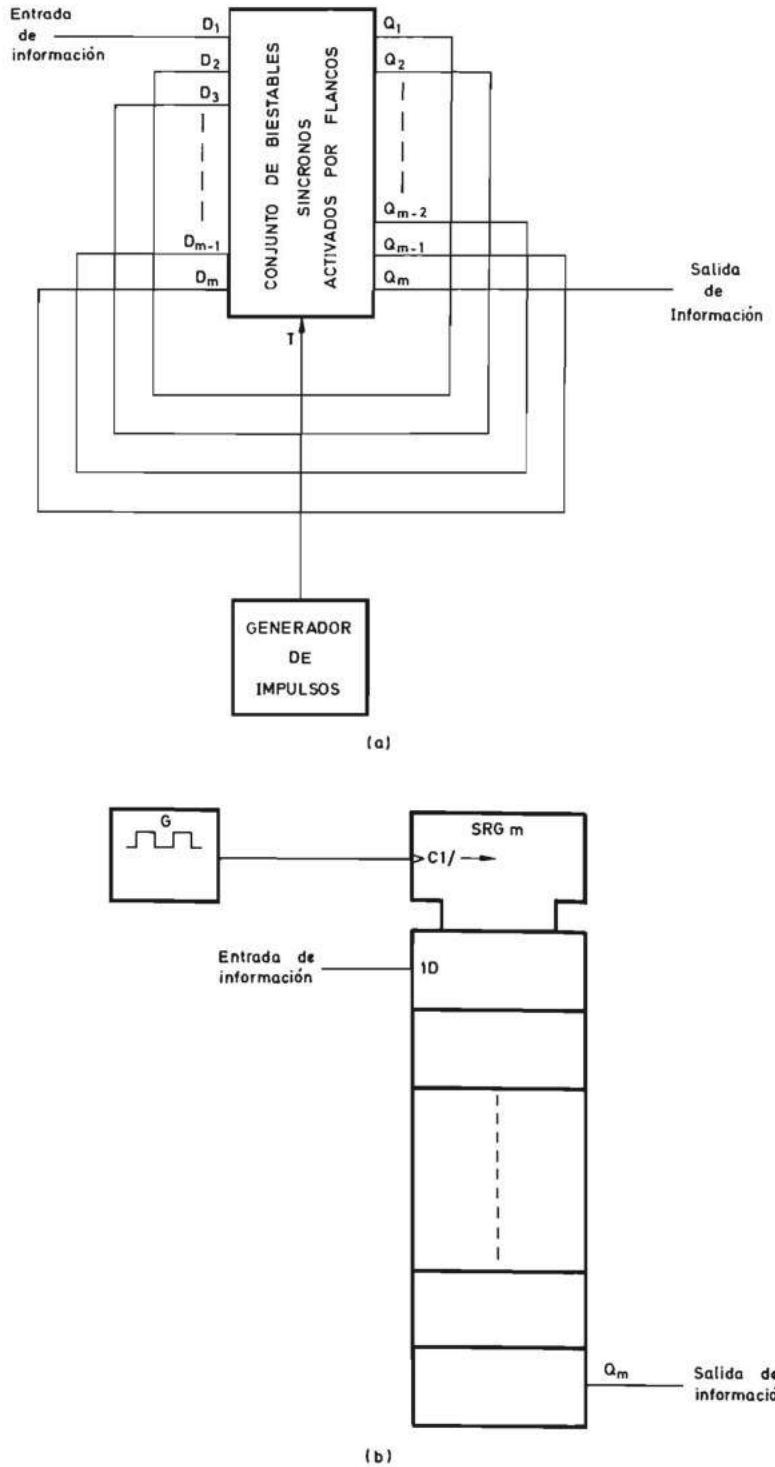


FIGURA 6.174.— **Diagrama de bloques de un registro de desplazamiento realizado con biestables *D*: *a*) Con indicación de todas las conexiones entre los biestables; *b*) Mediante un símbolo lógico normalizado.**

aplican a la entrada T de todos los biestables. Estos sistemas poseen la característica de que todos los biestables cambian simultáneamente.

Existe otra forma de construir los sistemas secuenciales síncronos que permite simplificar el sistema combinacional $N1$. Dicha forma consiste en conectar la entrada T de algunos biestables a la salida Q de otros y el generador de impulsos a los demás. La entrada T de un biestable se puede conectar a la salida Q de otro que cambie de estado como mínimo siempre que aquél deba cambiar de estado.

Estos sistemas presentan el inconveniente de que los biestables no cambian de estado simultáneamente y por ello se pueden generar transiciones aleatorias en las salidas del sistema $N2$ que genera las variables de salida Z.

Esta característica ha hecho que estos sistemas reciban el nombre de sistemas secuenciales síncronos asincronizados o asíncronos sincronizados, y que su utilización práctica presente limitaciones que han hecho que su principal aplicación sea la síntesis de contadores.

La comprensión del funcionamiento de estos sistemas resulta más sencilla mediante algunos ejemplos. En la figura 6.175 se representa el esquema general de un sistema secuencial síncrono asincronizado. La entrada T del primer biestable se

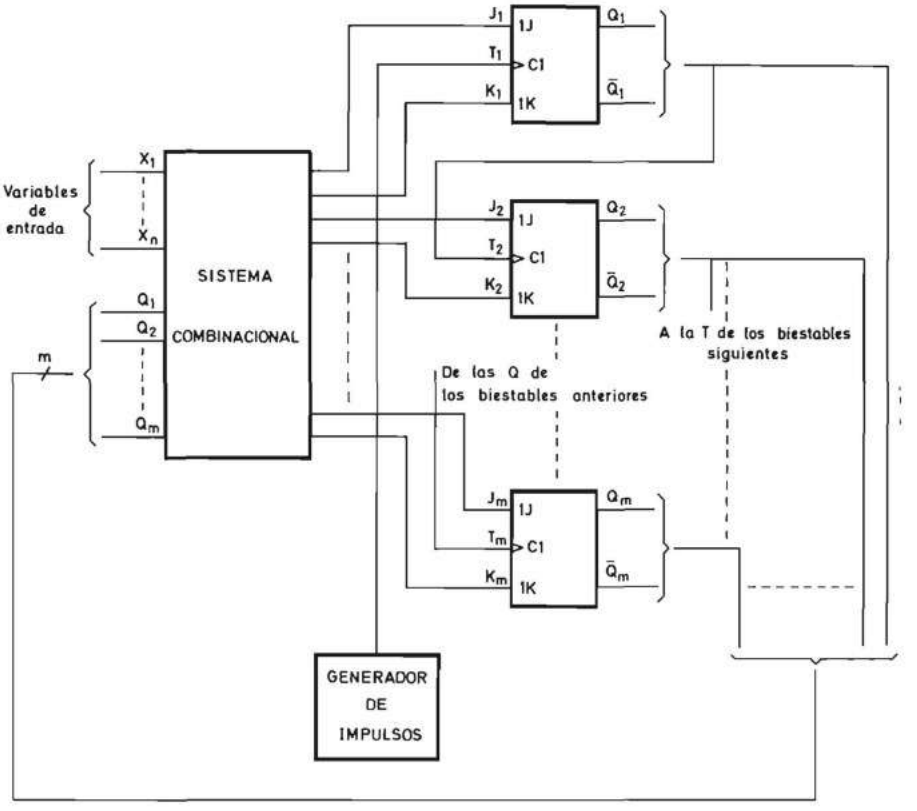


FIGURA 6.175.—**Diagrama de bloques de un sistema secuencial síncrono asincronizado.**

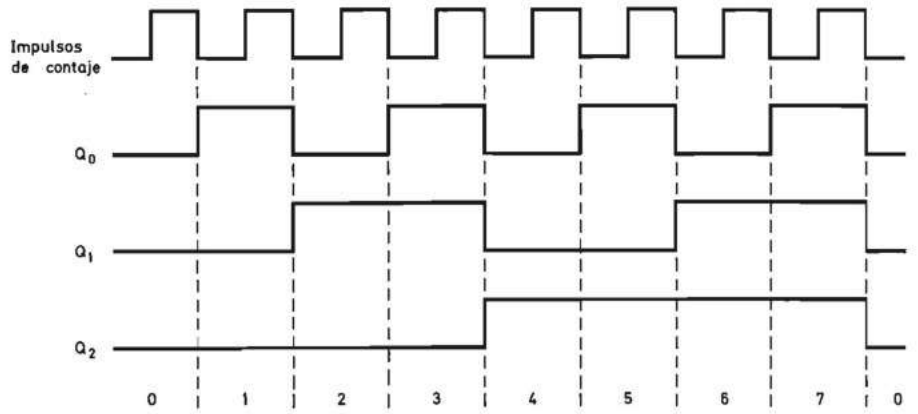


FIGURA 6.176.—**Diagrama de evolución temporal de las salidas de un contador asíncrono binario natural en sentido ascendente.**

conecta al generador de impulsos y la de los demás a las salidas Q de otros que les precedan. El biestable más adecuado para controlar la entrada T de otro es aquel que cambie de estado solamente cuando el último deba hacerlo. Si ello es posible para todos los biestables, se puede eliminar el sistema combinacional. Lo demostraremos mediante un ejemplo sencillo de gran aplicación práctica.

Supóngase que se desea realizar un contador asíncrono que efectúe el contaje de impulsos en binario natural y hasta 7. El diagrama de evolución de las variables de este contador se representa en la figura 6.176, en la que se supone que el cambio de los biestables se realiza en el flanco de bajada (se necesitan tres biestables para codificar los 8 estados posibles del 0 al 7).

Observando dicha figura se comprueba que Q_1 sólo debe cambiar cuando Q_0 lo hace de uno a cero y lo mismo puede decirse de Q_2 con respecto a Q_1. Por ello

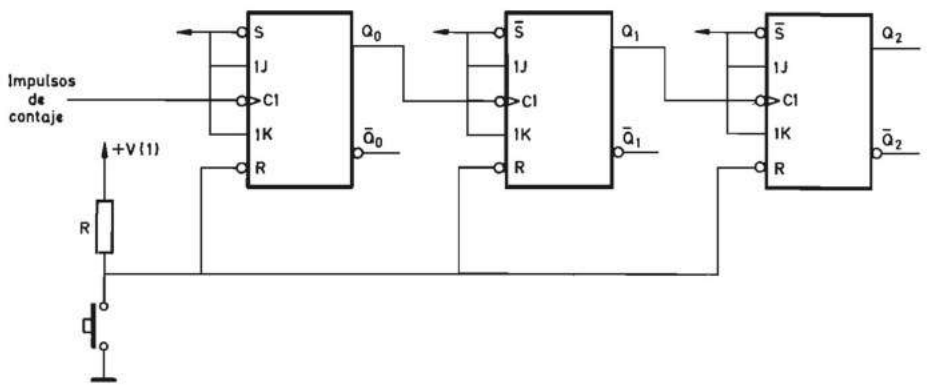


FIGURA 6.177.—**Esquema de un contador asíncrono binario natural en sentido ascendente.**

si el contador se realiza con biestables *J-K* activados por flancos de bajada se puede unir Q_0 a T_1 y Q_1 a T_2, al mismo tiempo que las entradas J y K de cada biestable se conectan al nivel del uno lógico. Se obtiene así el esquema del contador representado en la figura 6.177 en el que el sistema combinacional *N*1 ha desaparecido. En dicha figura se supone que los biestables tienen entradas de puesta a uno (*S*) y puesta a cero (*R*) activas con un cero. El contador es cíclico y evoluciona de acuerdo con el diagrama de flujo de la figura 6.178. Las entradas $\overline{R}$ se conectan a un interruptor que permite realizar una puesta a cero en cualquier instante.

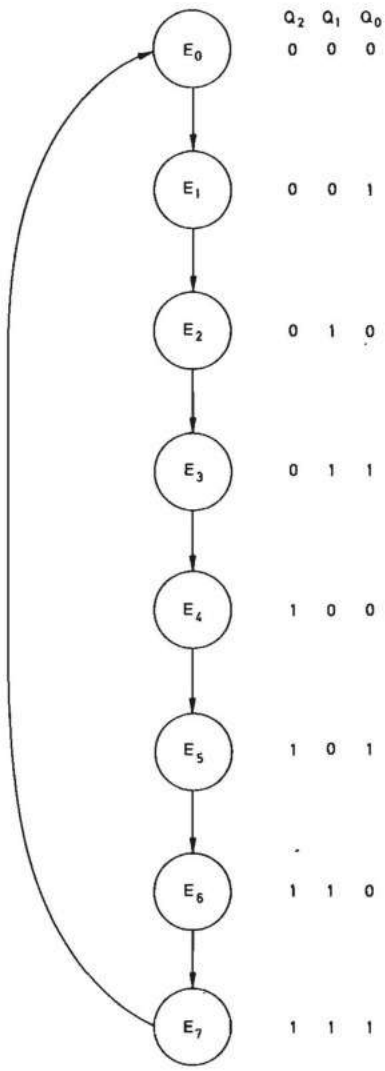


FIGURA 6.178.—**Diagrama de flujo de un contador asíncrono binario natural en sentido ascendente.**

El lector puede hacerse a continuación la siguiente pregunta: ¿Cuál sería el diagrama de flujo del contador si los biestables cambiasen de estado con los flancos de subida? Se deduce fácilmente que las salidas del contador evolucionarían de acuerdo con el diagrama de la figura 6.179 que corresponde al diagrama de flujo de la figura 6.180 equivalente al de un contador en binario natural en sentido descendente.

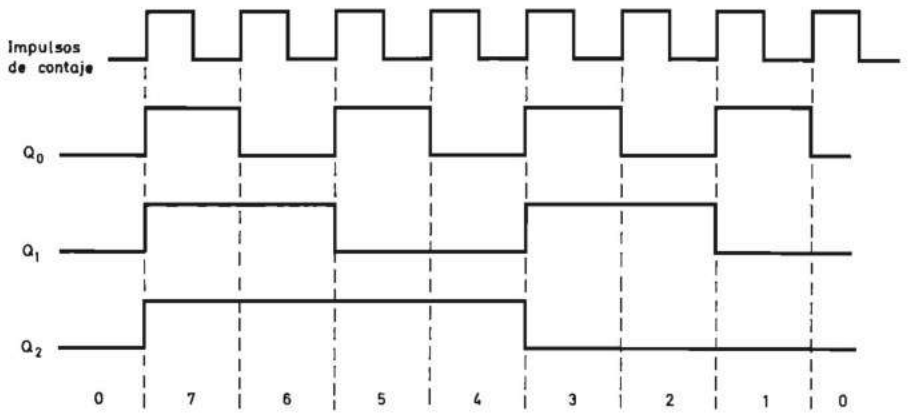


FIGURA 6.179.—**Diagrama de evolución temporal de las salidas de un contador asíncrono binario natural en sentido descendente.**

Otra pregunta interesante: ¿Cómo se podría obtener un contador descendente con un diagrama de flujo idéntico al de la figura 6.180 con biestables activados por flancos de bajada? La respuesta se da gráficamente en la figura 6.181 en la que se comprueba que $\overline{Q}_0$ se une a T_1 y $\overline{Q}_1$ a T_2.

Para elevar la capacidad de contaje es necesario añadir biestables conectándolos de forma similar (Q_2 o $\overline{Q}_2$ a T_3, etc.).

Se obtiene fácilmente un contador reversible combinando los esquemas de las figuras 6.177 y 6.181 mediante la utilización de multiplexores de dos canales junto con los biestables. El esquema se representa en la figura 6.182. Cada entrada T se conecta a la salida de un multiplexor de dos canales cuyas entradas se unen a las salidas Q y $\overline{Q}$ del biestable anterior. Las entradas de selección de todos los multiplexores se conectan a una variable $\overline{A}/D$ que según se encuentre en cero o uno hace que el contador realice el contaje en sentido ascendente o descendente respectivamente. En la figura 6.183 se representa el diagrama de flujo correspondiente.

Los contadores asíncronos se caracterizan por su sencillez y fueron los primeros en integrarse. Un estudio más detallado se realizará en el apartado 6.3.5.4.3.2 en el que se analizarán las transiciones debidas al retraso en la conmutación de los sucesivos biestables.

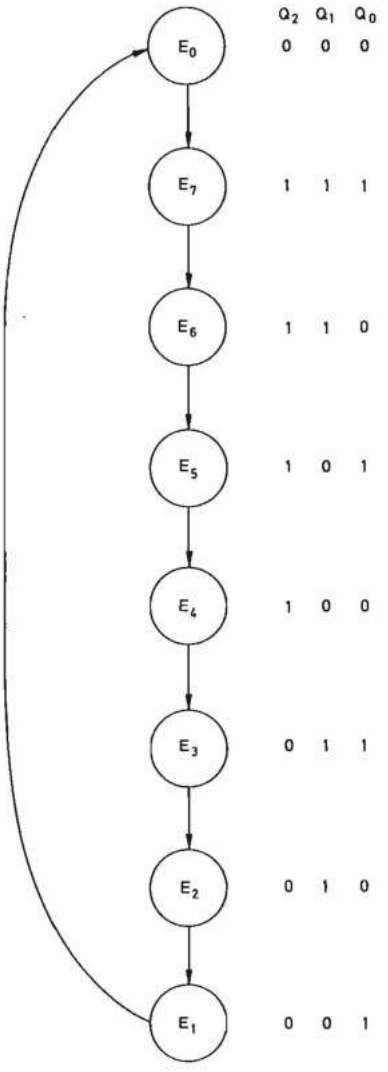


FIGURA 6.180.—**Diagrama de flujo de un contador binario natural en sentido descendente.**

6.3.5.4 Bloques funcionales síncronos

6.3.5.4.1 Introducción. En el apartado 6.3.5.2 se definieron y analizaron diversos sistemas secuenciales síncronos de aplicación general. Se define un bloque funcional síncrono, al igual que un bloque funcional combinacional, como un sistema secuencial síncrono de aplicación general.

Estos bloques funcionales poseen ya un nivel de complejidad lo suficientemente elevado como para que en ellos aparezcan diversos conceptos con variantes alternativas.

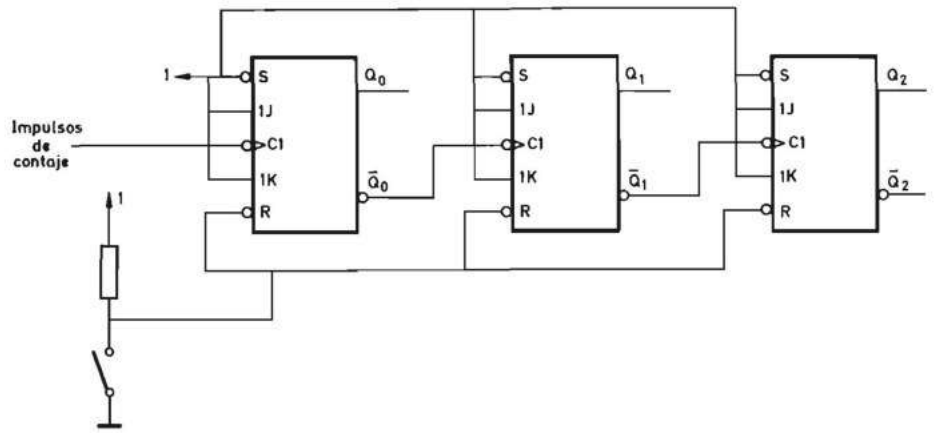


FIGURA 6.181.—**Esquema de un contador asíncrono binario natural en sentido descendente.**

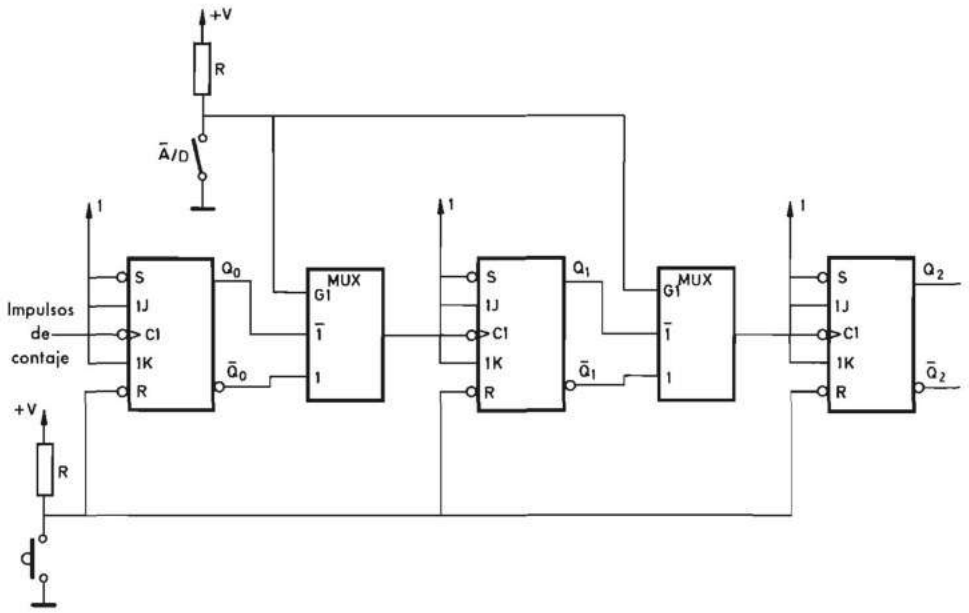


FIGURA 6.182.—**Esquema de un contador asíncrono reversible en binario natural.**

Esto hace necesario, en opinión del autor, la creación de una teoría general de los bloques funcionales síncronos que estudie los diferentes conceptos y sus variantes.

Por todo ello, se desarrolla en primer lugar dicha teoría para analizar a continuación los distintos tipos de bloques funcionales que, como se verá, son casos particulares de la misma.

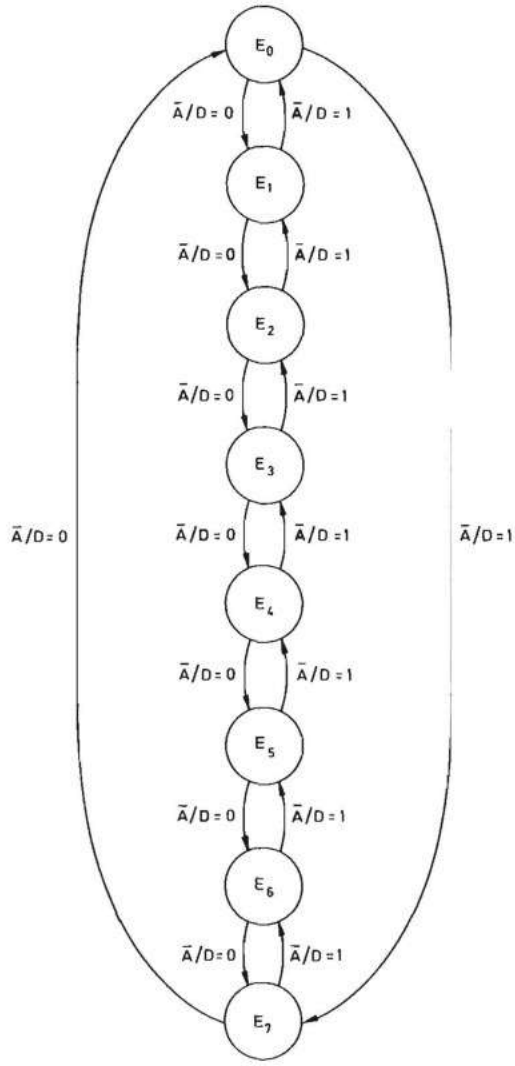


FIGURA 6.183.— **Diagrama de flujo de un contador reversible en binario natural.**

Se recomienda al lector que antes de estudiar los apartados siguientes analice con detenimiento la simbología normalizada descrita en el apéndice 1.

6.3.5.4.2 Teoría general y símbolos normalizados. La creación de una teoría general de los bloques funcionales síncronos está basada en la existencia de un conjunto de características comunes a todos ellos independientemente de la función concreta que realicen.

Los bloques funcionales síncronos poseen en general una capacidad de operación

(por ejemplo contaje, memorización, etc.) limitada que depende del número de biestables situados en su interior.

Las características generales comunes a todos los bloques funcionales síncronos son las siguientes:

a) Posibilidad de puesta en uno o más estados determinados de todos los biestables que forman el bloque.

Esta característica responde a la necesidad de colocar los biestables en un cierto estado en el momento de dar tensión al sistema electrónico del cual forma parte el bloque, o cuando se produce una determinada circunstancia.

La puesta en un cierto estado se puede realizar de forma asíncrona o síncrona. La puesta asíncrona en cierto estado actúa directamente sobre las entradas de puesta a cero o puesta a uno asíncronas de los biestables. La más usual en los bloques funcionales síncronos es la puesta a cero de todos los biestables.

La puesta síncrona en un cierto estado se logra colocando las entradas de información de todos los biestables en un estado tal que al aplicar un flanco activo en la entrada T se pongan aquéllos en un cierto estado.

Por ejemplo, si se trata de una puesta a cero síncrona y los biestables son del tipo *J-K*, se pondrán a cero todas las entradas J y a uno todas las entradas K.

La más utilizada de ambas es la puesta a cero asíncrona que la poseen la práctica totalidad de los bloques funcionales síncronos en las distintas tecnologías.

b) Selección de distintos modos de operación.

La mayoría de los diferentes tipos de bloques funcionales pueden tener varios modos de operación, algunos de los cuales se analizaron en el apartado 6.3.5.2.

Los modos de operación dependen, en general, del tipo de bloque. Por ejemplo, los contadores pueden realizar el contaje en dos sentidos diferentes seleccionables mediante una variable binaria de selección (variable A/D en la figura 6.167).

En los esquemas es conveniente que se pueda conocer el modo de operación correspondiente a cada uno de los estados de la variable de selección. Una forma de realizarlo, utilizada en la simbología no normalizada, consiste en definir la variable de selección mediante las letras iniciales de los dos modos de operación, separadas por un segmento inclinado. Además, a la letra correspondiente al modo de operación que se produce cuando la variable de selección se encuentra en cero, se le coloca encima la raya de inversión. Por ejemplo, en el contador reversible de la figura 6.167 la variable de selección se indica $\overline{A}/D$ si el contador cuenta en sentido ascendente cuando $A/D = 0$ y en sentido descendente cuando $A/D = 1$. La representación $A/\overline{D}$ indicará la situación contraria a la anterior.

Un modo de operación, de aplicación en todos los bloques funcionales síncronos independientemente de su tipo, es el de inhibición. La inhibición de un bloque funcional síncrono consiste en colocar las entradas de los biestables en un estado tal que los impulsos aplicados a la entrada de sincronización T no producen la modificación del estado interno. Por ejemplo, si los biestables utilizados son *J-K*, se logra la inhibición colocando en nivel cero las entradas J y K de todos ellos.

La inhibición produce dos modos o estados de funcionamiento: estado de inhibición y estado de desinhibición. Ambos pueden ser, por lo tanto, seleccionados

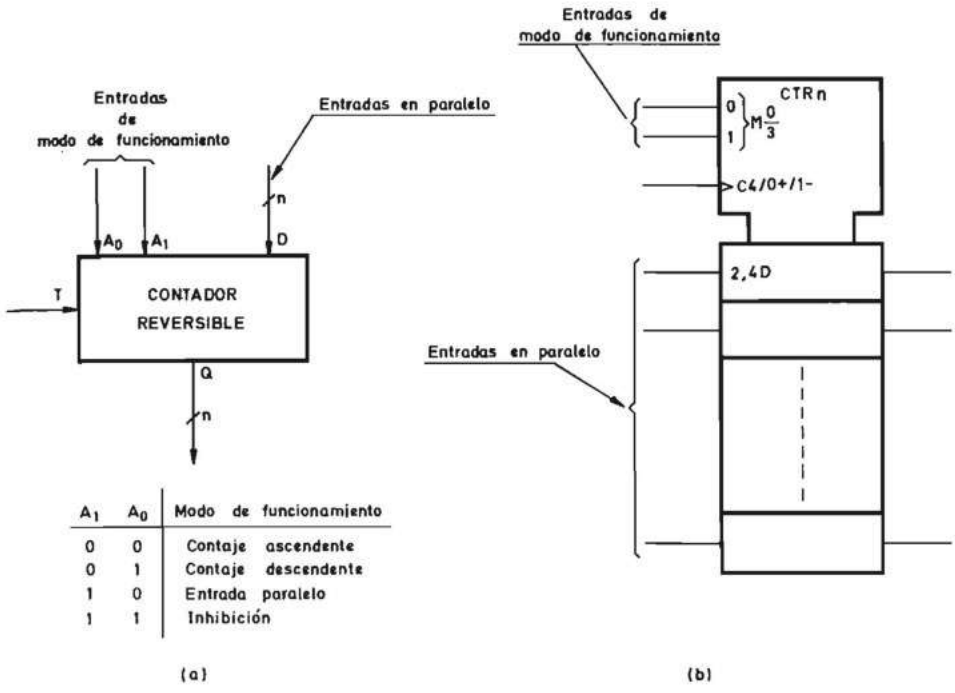


FIGURA 6.184.— **Símbolos lógicos de un contador síncrono reversible: *a)* No normalizado; *b)* Normalizado.**

mediante una variable binaria que se suele designar mediante la letra I si la inhibición se produce cuando adopta el nivel uno y por $\bar{I}$ en el caso contrario.

Cuando los distintos modos de funcionamiento son incompatibles entre sí, es decir, que el sistema sólo se puede encontrar en uno de ellos, y no en más de uno simultáneamente, son posibles varias soluciones para las entradas de selección de aquéllos.

Un ejemplo se muestra en las figuras 6.184 y 6.185 en las que se representa el diagrama de bloques de un contador reversible con cuatro modos de funcionamiento. En la figura 6.184 se presenta una solución en la que se utilizan dos entradas de selección A_0 y A_1 que permiten mediante sus cuatro posibles estados seleccionar los cuatro modos: contaje ascendente, contaje descendente, entrada de información en paralelo e inhibición.

En la figura 6.184*a* se representa el símbolo no normalizado. Una característica de este símbolo es que no es autoexplicativo y necesita ir acompañado de una tabla de verdad que indique el modo correspondiente a cada una de las combinaciones de las variables de selección A_0 y A_1. Dicha tabla se representa debajo del símbolo en la figura 6.184*a*. La figura 6.184*b* representa el símbolo normalizado que hace innecesaria la tabla de verdad. Para ello a las variables de selección se les asigna la letra M (de modo de operación) seguida de 0/3, que indica que las combi-

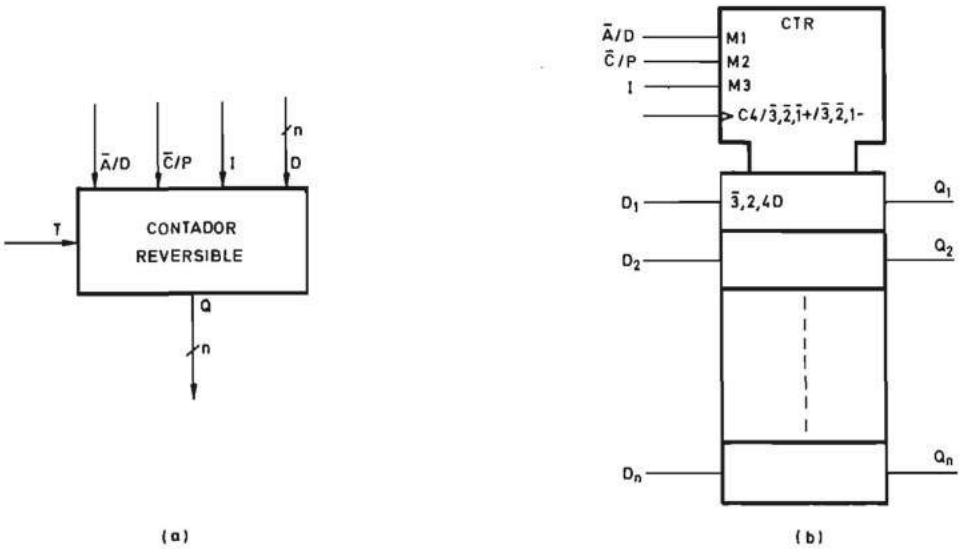


FIGURA 6.185.— **Símbolos lógicos de un contador reversible: *a*) No normalizado; *b*) Normalizado.**

naciones 0, 1, 2 y 3 corresponden a los distintos modos de operación. La combinación 0 corresponde al contaje ascendente, lo cual se indica colocando los símbolos 0+ en la entrada de contaje. La combinación 1 corresponde al contaje descendente, lo cual se indica colocando los símbolos 1— en la entrada de contaje. La combinación 2 permite la entrada de información en paralelo, lo cual se indica colocando el dígito 2 en las entradas en paralelo. Además, dado que la entrada en paralelo se produce en sincronismo con los impulsos aplicados en la entrada de impulsos, ésta se denomima *C* (control) seguida del número 4.

La entrada en paralelo recibe, por lo tanto, la denominación 2,4*D* que indica que para que se produzca es necesario que *M* 0/3 = 2 y que simultáneamente se aplique un flanco activo (en este caso el de subida) en *C*4. La entrada de control recibe la denominación *C*4/0 + /1 — en la que las barras inclinadas indican modos de funcionamiento excluyentes. Se recomienda al lector nuevamente que consulte el apéndice 1 en el caso de que no conozca la simbología normalizada.

La figura 6.185*a* representa el símbolo no normalizado de otra solución que posee tres entradas de selección, una de ellas $\overline{A}/D$ selecciona si el contaje se realiza en sentido ascendente o descendente. La entrada $\overline{C}/P$ selecciona si el impulso aplicado en *T* es contado o bien hace que entre en paralelo la información presente en las entradas *D*. Finalmente, la entrada *I* inhibe cualquier acción cuando se encuentra en nivel uno. Esta solución implica un terminal más en el circuito integrado, pero presenta la ventaja de que es autoexplicativa y no necesita la tabla de la figura 6.184*a*. Para ello se utiliza el convenio, indicado anteriormente, de representar las entradas de selección de modo de operación por las dos iniciales de los dos modos seleccionados, separadas por un segmento inclinado.

En la figura 6.185*b* se representa el símbolo normalizado de esta solución. Las tres entradas de modo de operación se denominan *M*1 ($\overline{A}/D$), *M*2 ($\overline{C}/P$) y *M*3 (*I*). La entrada de impulsos de sincronismo recibe la denominación *C*4/$\overline{3}$, $\overline{2}$, $\overline{1}$ +/$\overline{3}$, $\overline{2}$, 1— que indica:

*C*4: impulsos de entradas de información.

$\overline{3}$, $\overline{2}$, 1+: contaje ascendente (+) si *M*3 = 0 ($\overline{3}$), *M*2 = 0 ($\overline{2}$) y *M*1 = 0

$\overline{3}$, $\overline{2}$, 1—: contaje descendente (—) si *M*3 = 0 ($\overline{3}$), *M*2 = 0 ($\overline{2}$) y *M*1 (1)

La entrada de información en paralelo recibe la denominación $\overline{3}$, 2, 4*D* que indica que dicha acción se produce si *M*3 = 0 ($\overline{3}$), *M*2 = 1 (2) y se aplica un flanco de subida en *C*4 (4*D*).

c) Ampliabilidad de la capacidad.

Los bloques funcionales integrados poseen una capacidad limitada, pero es conveniente que se puedan acoplar para ampliar dicha capacidad sin necesidad de utilizar otro tipo de circuitos.

Para ello se dota a los bloques funcionales síncronos de los terminales adecuados. El modo de operación de estos terminales depende del tipo de bloque funcional y se estudia en los apartados 6.3.5.4.3 y siguientes dedicados a cada uno de ellos.

d) Control del tercer estado de la salida.

Ciertos tipos de bloques funcionales se utilizan en ocasiones conectándolos a una barra (bus) de información. Es conveniente para ello que tengan en las salidas puertas seguidoras de tres estados como las estudiadas en el apartado 5.4.4.3.4.

En la figura 6.186*a* se representa el esquema simplificado de un bloque funcional síncrono con salida de tres estados. Las salidas *Q* de los biestables del bloque se conectan a las entradas de sendas puertas de tres estados cuya entrada de inhibición se lleva al exterior. Según dicha entrada se encuentre en el nivel lógico cero o uno, a la salida del bloque aparece el estado de las salidas *Q* de los biestables, o bien dicha salida se pone en el tercer estado de alta impedancia con relación a los terminales de la fuente de alimentación. En la figura 6.186*b* se representa el símbolo lógico no normalizado, en el que la salida de tres estados se indica con un triángulo invertido.

En la figura 6.186*c* se representa el símbolo normalizado. La entrada de inhibición recibe la denominación *EN* [desinhibición (enable)] seguida del dígito α. Dicho dígito α se coloca al lado del símbolo ∇ de tres estados en las salidas. En el caso de que *EN* α actúe sobre todas las salidas, es suficiente que se coloque $\alpha\nabla$ en una sola de ellas. Cuando el estado de la salida depende sólo de *EN* se puede incluso suprimir α.

e) Otras funciones.

Las funciones descritas en los apartados anteriores son las más usuales, pero existen otras que, aunque de uso menos generalizado, son de utilidad en algunos bloques funcionales.

En la figura 6.187*a* se representa el símbolo no normalizado de una de ellas constituida por la posibilidad de selección del flanco activo de los impulsos de sin-

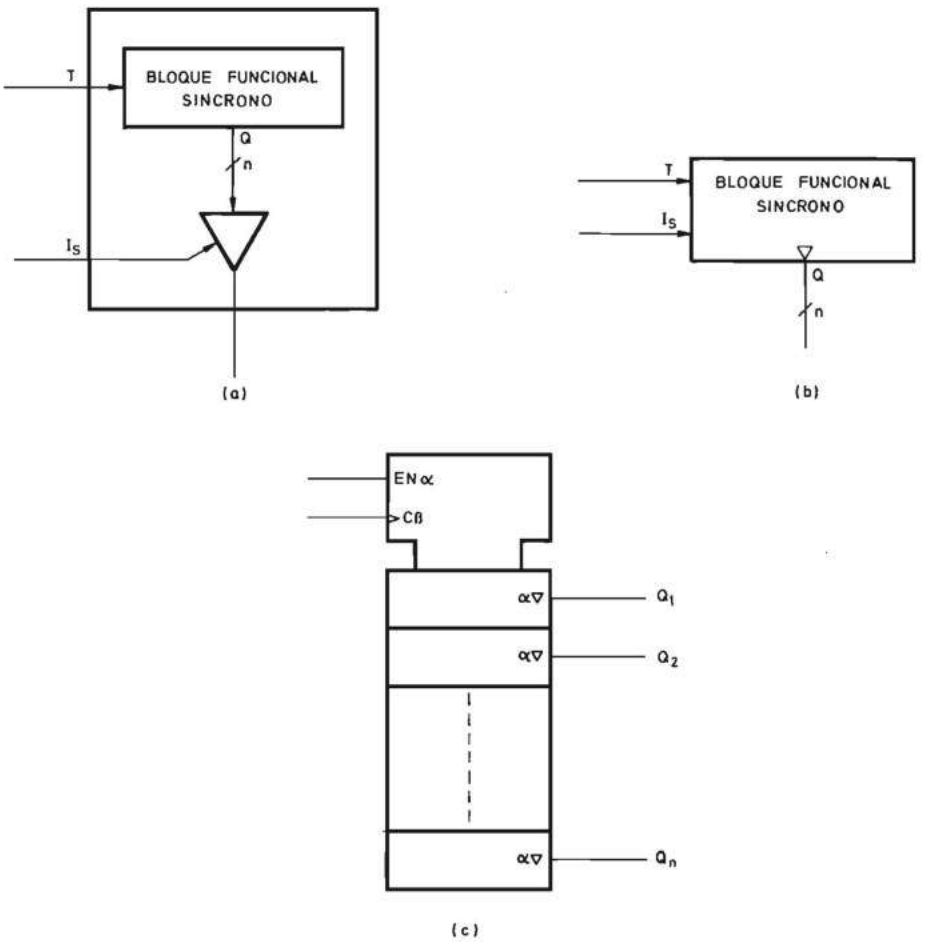


FIGURA 6.186.—**Bloque funcional síncrono con salida de tres estados: *a*) Esquema simplificado; *b*) Símbolo lógico no normalizado; *c*) Símbolo lógico normalizado.**

cronismo. La forma mejor de lograrlo es conectar la entrada T del bloque a una puerta O-exclusiva de dos entradas que se disponen en el exterior. A una de ellas se aplican los impulsos de sincronismo. Según que el nivel de la otra sea cero o uno, a la salida de la puerta aparecen los impulsos de forma directa o inversa. Por ello, esta entrada se puede denominar de selección de flanco activo y ser indicada por $\overline{S}/B$ si su estado cero selecciona el flanco activo de subida y el uno el de bajada, o por $S/\overline{B}$ en caso contrario. En la figura 6.187*b* se representa el correspondiente símbolo normalizado. La entrada $\overline{S}/B$ posee el indicativo $M1$. La entrada de impulsos de sincronismo se indica mediante dos terminales, uno activo con los flancos de subida ($\overline{1}C$) y otro con los de bajada ($1C$). El dígito $\overline{1}$ indica que la acción correspondiente se produce cuando $M1 = 0$, y el dígito 1 indica que la acción correspondiente se produce cuando $M1 = 1$.

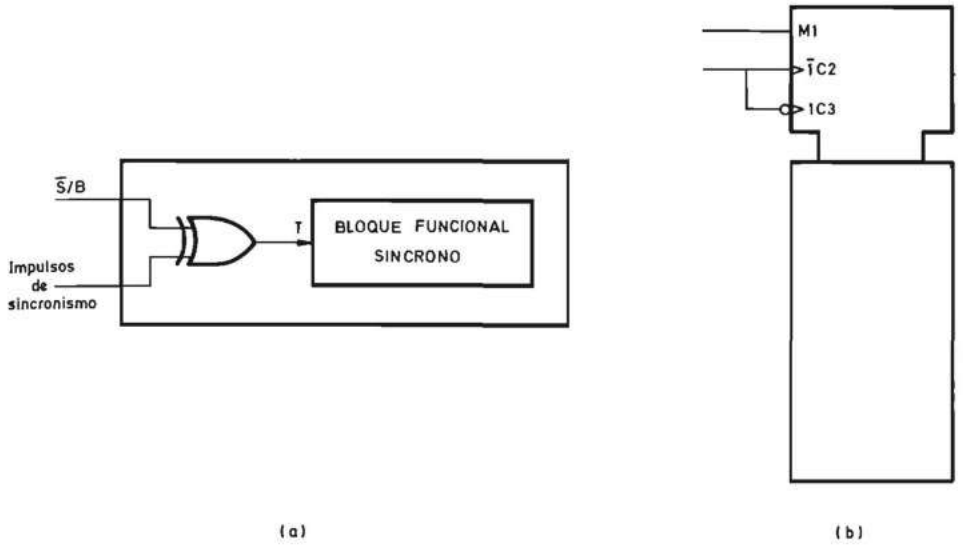


FIGURA 6.187.—**Bloque funcional síncrono con selección del flanco activo de los impulsos de sincronismo: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**

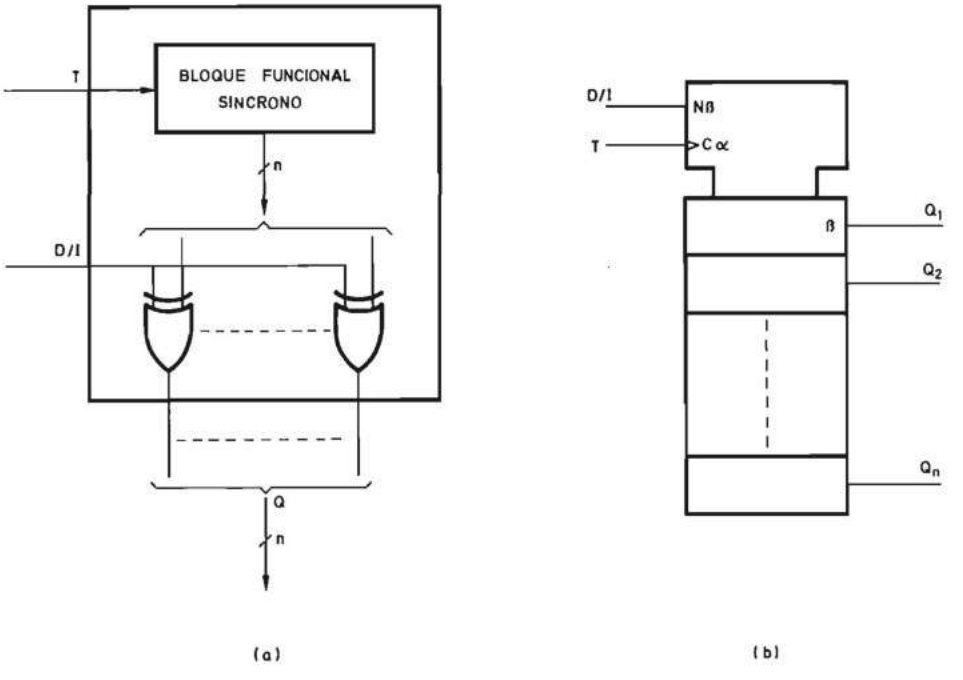


FIGURA 6.188.—**Bloque funcional síncrono con selección de salida directa o inversa: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**

En la figura 6.188*a* se representa el símbolo no normalizado de otra función constituida por la posibilidad de selección de que a la salida del circuito aparezca el contenido de los biestables que forman parte del bloque funcional o su inverso. Para ello las salidas de los n biestables se conectan a una de las entradas de otras tantas puertas O-exclusiva de dos entradas. Las segundas entradas de todas las puertas se unen entre sí y al exterior a un terminal que, según se encuentre en cero o en uno, produce el efecto indicado anteriormente. Dicho terminal recibe la denominación D/I (directo/inverso) y, de acuerdo con lo expuesto anteriormente, se debe indicar $\overline{D}/I$ o $D/\overline{I}$. En la figura 6.188*b* se representa el símbolo normalizado.

La entrada D/I recibe la denominación N [de inversión (negate)] seguido del dígito β. Las salidas llevan el indicativo β que indica que son directas si $N\beta = 0$ e inversas si $N\beta = 1$.

Como resumen, en la figura 6.189 se representan los símbolos no normalizado (*a*) y normalizado (*b*) de un bloque funcional síncrono que posee todos los terminales estudiados en este apartado y los que le preceden. Es conveniente describir brevemente el símbolo normalizado que posee los siguientes terminales:

$M \frac{0}{2^m - 1}$: m entradas de modo de operación que corresponden a 2^m modos diferentes.

Gk : Entrada de desinhibición que actúa sobre la salida de propagación k $CT = n$.

$M\alpha$: Entrada de control del tipo de flanco activo.

$EN\lambda$: Entrada de desinhibición del tercer estado de la salida.

$N\gamma$: Entrada de control de la inversión de la salida.

$\alpha C1$: Entrada de impulsos de sincronismo activa con flancos de subida cuando $M\alpha = 1$

$\overline{\alpha}C1$: Entrada de impulsos de sincronismo activa con flancos de bajada cuando $M\alpha = 0$

$1D$: Entrada de información en paralelo en sincronismo con los impulsos aplicados en $\alpha C1$ y $\overline{\alpha}C1$.

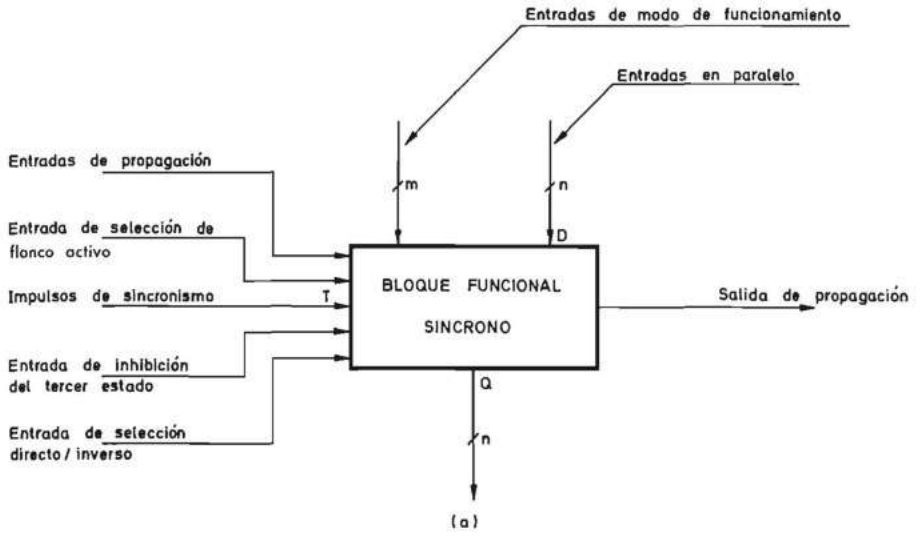
$\lambda,\gamma\ \nabla$: Salida de tres estados. Se encuentra en tercer estado si $EN\lambda = 0$. Cuando $EN\lambda = 1$, aparece en ella la información contenida en el bloque si $N\gamma = 0$ y su inversa si $N\gamma = 1$.

$kCT = n$: Salida de propagación que se encuentra en nivel uno si $Gk = 1$ y el contenido del bloque es el número binario equivalente al decimal n.

f) Programabilidad.

Finalmente se estudia esta característica que añade una gran flexibilidad a los bloques funcionales síncronos. Se dice que un bloque funcionl síncrono es programable cuando posee capacidad de memorización de las variables de modo de operación.

En la figura 6.190*a* y *b* se representa el esquema básico de un bloque funcional síncrono programable que está dotado de un registro de entrada y salida en paralelo cuyas salidas se conectan a las entradas de modo de operación del bloque funcional síncrono. Colocando en las entradas D de este registro una combinación bi-



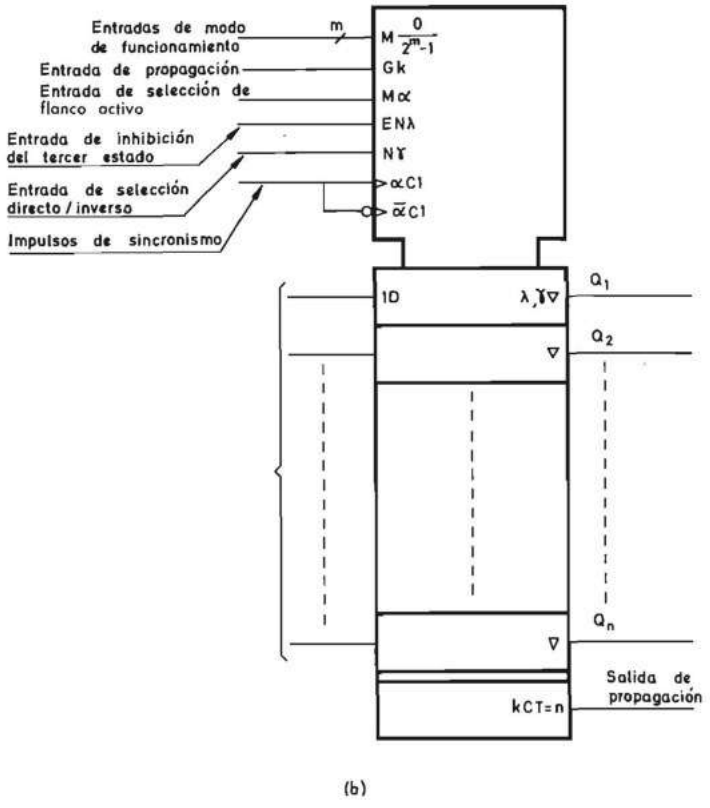


FIGURA 6.189.—**Diagrama de bloques general de un bloque funcional síncrono: *a)* Símbolo lógico no normalizado; *b)* Símbolo lógico normalizado.**

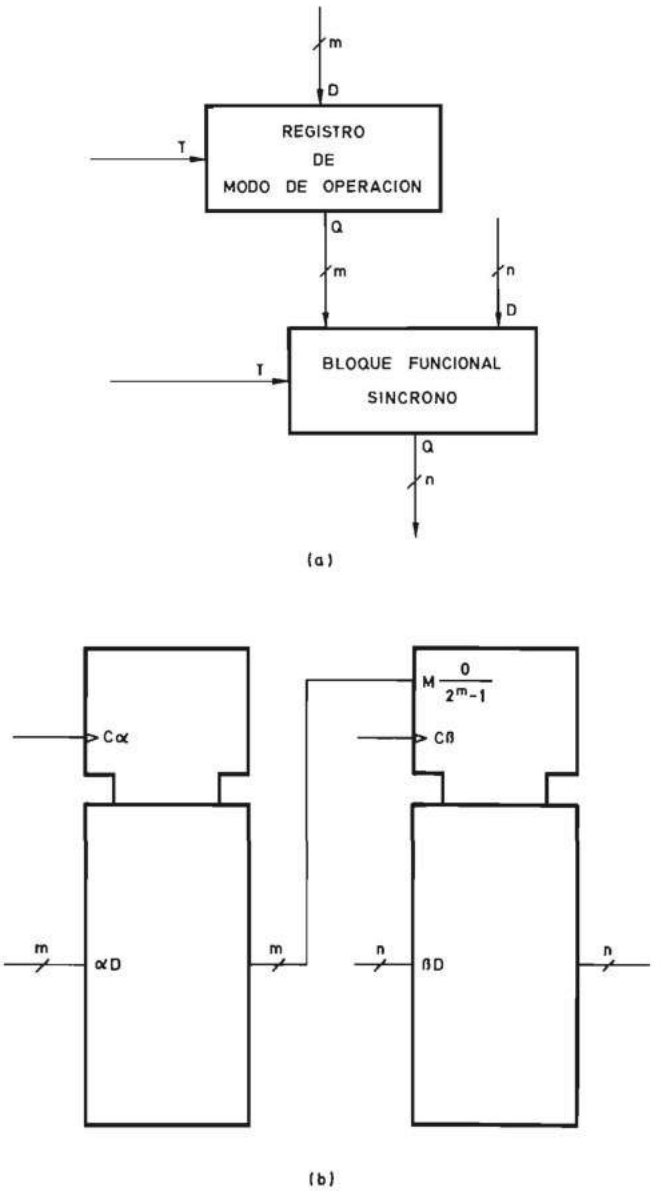
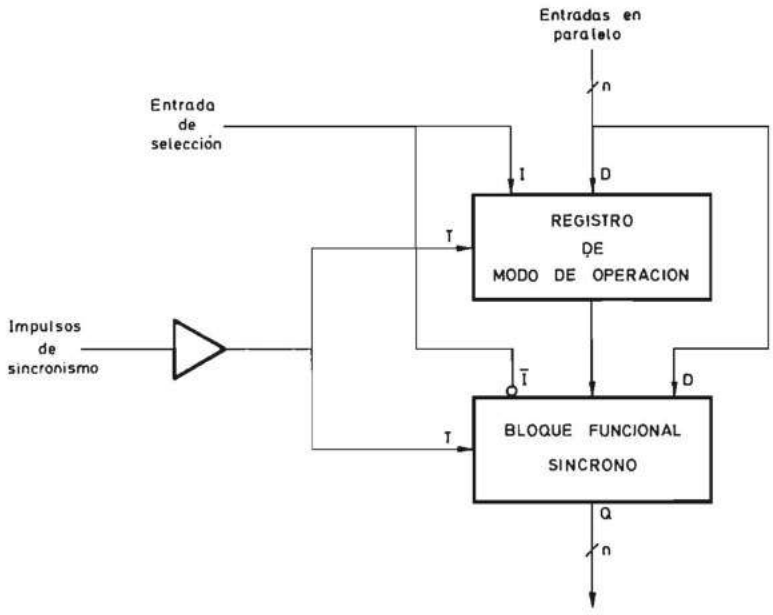


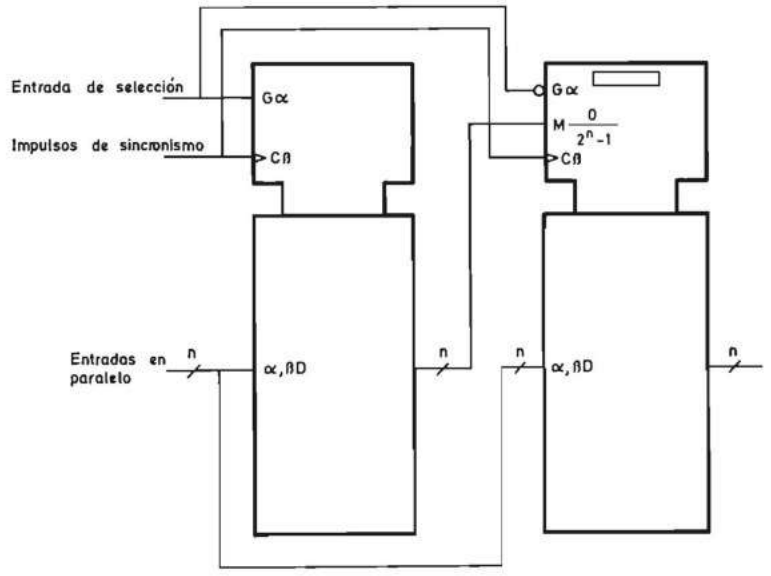
FIGURA 6.190.—**Esquema básico de un bloque funcional síncrono programable realizado mediante: *a*) Símbolos lógicos no normalizados; *b*) Símbolos lógicos normalizados.**

naria y aplicando un impulso en su entrada $T(C)$ se memoriza en él un cierto modo de funcionamiento.

En la figura 6.190 se supone que las entradas en paralelo del registro y del bloque funcional síncrono son independientes al igual que las entradas $T(C)$ de am-



(a)



(b)

FIGURA 6.191.—**Esquema básico de un bloque funcional síncrono programable con entrada en paralelo única realizado mediante: *a*) Símbolos lógicos no normalizados; *b*) Símbolos lógicos normalizados.**

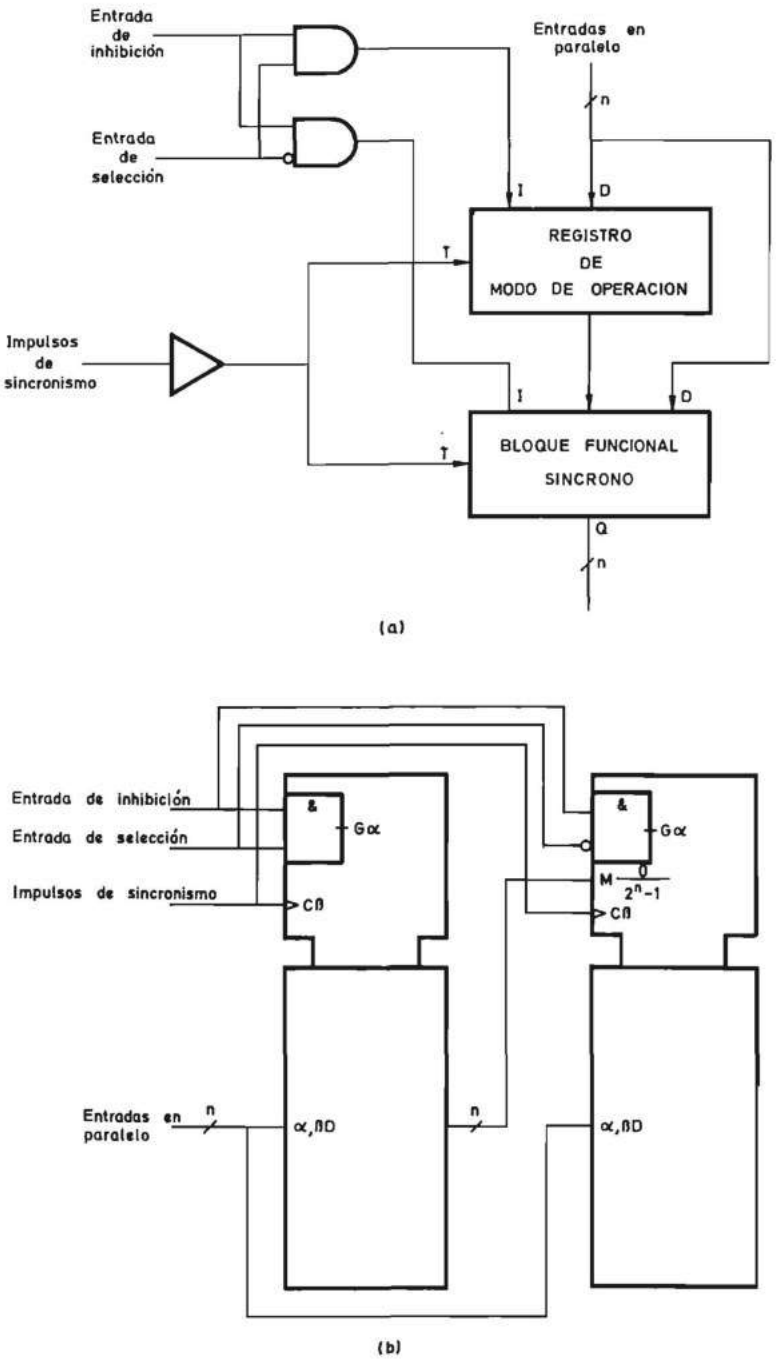


FIGURA 6.192.—**Esquema básico de un bloque funcional síncrono programable con entrada en paralelo única y entrada de inhibición realizado mediante: *a*) Símbolos lógicos no normalizados; *b*) Símbolos lógicos normalizados.**

bos. Pero, en general, es necesario seleccionar primero el modo de operación para realizar a continuación las acciones adecuadas sobre el bloque funcional síncrono, por ello resulta más adecuado disponer de terminales comunes de entrada en paralelo y de una entrada $T(C)$ también común. En la figura 6.191*a* y *b* se representa el esquema básico. En el registro se memorizan todas las señales de modo de operación excepto la de inhibición. Las entradas $T(C)$ de ambos elementos se conectan entre sí, y lo mismo sucede con las entradas en paralelo cuyo número es igual a n o m según cuál de ambos sea mayor. Colocando una cierta información en los n bits de entrada en paralelo, al aplicar un impulso en la entrada $T(C)$, dicha información se introduce en el registro o en el bloque funcional dependiendo del estado de la entrada de inhibición/desinhibición de ambos. Dado que la información debe entrar en uno o en otro, se puede dotar a ambos de sendos terminales de inhibición/desinhibición $I(G\alpha)$ que actúen con niveles inversos y unirlos entre sí. Ese terminal común constituye una entrada de selección. En la figura 6.191, si ese terminal se encuentra en nivel uno, la información entra en el registro de modo de operación, y si se encuentra en nivel cero, entra en el bloque funcional.

Se puede además dotar al bloque funcional obtenido de una entrada de inhibición común a ambos elementos. En la figura 6.192*a* y *b* se realiza esta función mediante sendas puertas Y conectadas a las entradas de inhibición. Sólo cuando la entrada de inhibición se encuentra en nivel uno puede introducirse información en paralelo en el registro o en el bloque funcional. En la figura 6.193*a* y *b* se representa un bloque funcional síncrono programable que contiene en su interior todos los elementos de la figura 6.192.

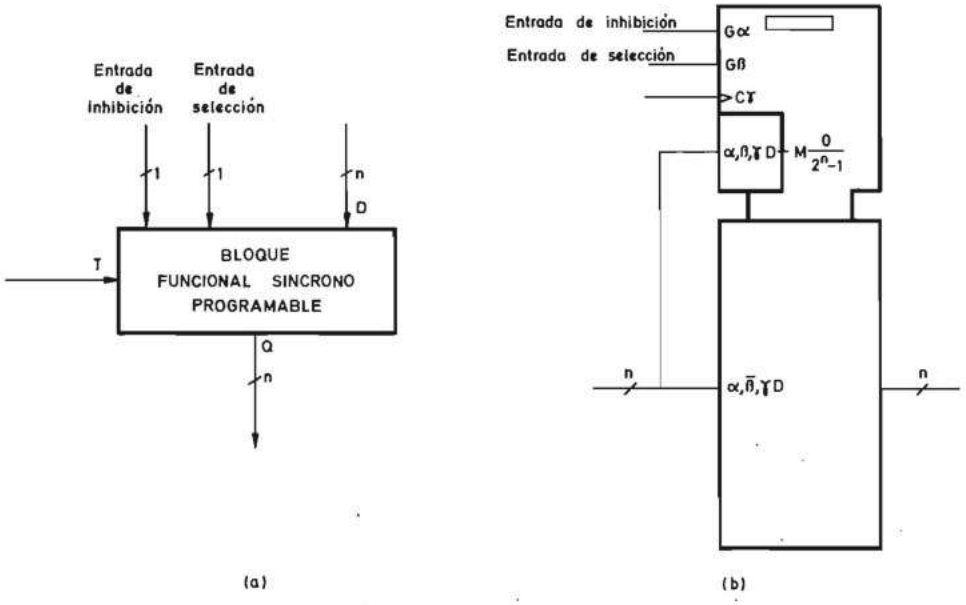


FIGURA 6.193.—**Bloque funcional síncrono programable: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**

En apartados sucesivos se estudian los bloques funcionales síncronos más típicos que se pueden considerar casos particulares de la teoría general que se acaba de exponer.

6.3.5.4.3 Contadores

6.3.5.4.3.1 Generalidades. Como ya se ha visto anteriormente, los contadores son sistemas secuenciales con una entrada de impulsos, cuyo estado interno en cada instante representa el número de impulsos que se han aplicado a aquélla. El número de estados internos ha de ser igual al máximo número de impulsos que se desean contar, más uno que será el estado inicial, el cual indicará la ausencia de impulsos.

Para la realización de los contadores se utilizan los elementos biestables sincronizados por flancos estudiados en el apartado 6.3.2 que poseen dos estados internos.

Los parámetros más importantes de los contadores son los siguientes:

1. Frecuencia máxima de los impulsos que pueden contar.

Depende de la tecnología utilizada en la realización de los biestables y las puertas lógicas que constituyen el contador. Dicha tecnología fija el tiempo de conmutación de aquéllos y el tiempo de propagación de éstas. Estos parámetros han tenido una gran evolución y, para conocer en cada momento los valores reales, es necesario recurrir a los manuales de los fabricantes.

2. Código de contaje.

El código binario de contaje de impulsos puede ser cualquiera de los muchos existentes y algunos de los cuales se estudiaron en el capítulo 1. Los más utilizados son el binario natural y el decimal codificado en binario natural (BCD natural).

Cuando la capacidad de integración de los fabricantes sólo había alcanzado el nivel de puerta o biestable era necesario realizar el diseño de los contadores mediante interconexiones de los citados elementos. El desarrollo de las técnicas de integración ha permitido la fabricación de circuitos integrados monolíticos complejos que contienen contadores que realizan el contaje en los códigos anteriormente citados. Al lector que desee analizar los métodos de diseño de contadores con biestables se le remite a la bibliografía al final del capítulo [MAND 83].

3. Capacidad de contaje.

Es el mayor número de impulsos que han de ser contados por el sistema. Este parámetro fija el número de elementos biestables que se han de utilizar. Si llamamos n al número de impulsos que se han de contar, el número de biestables necesarios m, ha de cumplir la relación:

$$2^{m-1} < n + 1 \leq 2^m$$

Por ejemplo, si el número de impulsos a contar es 9, el contador ha de tener 4 biestables, dado que

$$2^{4-1} < 9 + 1 \leq 2^4$$

4. Modo de operación.

Los contadores pueden ser síncronos o asíncronos según su modo de operación

corresponda a un sistema secuencial síncrono o síncrono asincronizado, respectivamente.

Los contadores síncronos son aquellos en los que todas las variables de estado cambian simultáneamente, para lo cual se aplican los impulsos de contaje a la entrada T de todos los biestables. La máxima frecuencia de operación de estos contadores es igual a la de los biestables utilizados en su realización.

Los contadores asíncronos son aquellos en los que las variables de estado interno no cambian simultáneamente. Los impulsos que se desean contar no se aplican a la entrada T de todos los biestables sino, en general, sólo a la del primero, y la entrada T de los demás biestables es gobernada por las salidas de los que les preceden. Por tanto, para cambiar de un estado interno a otro no adyacente cuando se aplica un impulso al contador, éste pasa por una serie de estados intermedios.

Las principales diferencias entre los contadores síncronos y asíncronos son las siguientes:

a) La máxima frecuencia de operación de los contadores síncronos es superior a la de los asíncronos porque, antes de aplicar a la entrada de éstos un nuevo impulso, es necesario que todos los biestables hayan cambiado de estado.

b) Los contadores asíncronos son menos complejos que los síncronos porque el número de puertas que se utilizan para controlar las entradas de los biestables es menor.

c) Los contadores asíncronos presentan el problema de la aparición de impulsos aleatorios en la decodificación de sus estados.

En sucesivos apartados se analizan los bloques funcionales contadores tanto asíncronos como síncronos.

6.3.5.4.3.2 Contadores asíncronos. Los bloques funcionales contadores asíncronos responden a la teoría desarrollada en el apartado 6.3.5.3. Fueron los primeros en ser integrados porque son los que necesitan un menor número de puertas.

El diagrama de bloques de un bloque funcional contador asíncrono se representa en la figura 6.194*a* y *b* y posee los siguientes terminales:

a) Entrada T (C) de impulsos de contaje (se supone en este caso activa con flancos de bajada).

b) Entradas de puesta en estado inicial que actúan por nivel e independientemente de los impulsos de contaje. La más usual es la de puesta a estado cero.

c) Salidas Q de los biestables.

Los primeros en ser integrados fueron bloques funcionales de 4 biestables con capacidad máxima de contaje 9 en BCD natural o 15 en binario natural.

Los bloques funcionales contadores asíncronos en binario natural o BCD natural se diferencian sólo por el esquema del circuito combinacional que hace que se vuelva al estado cero a partir del 9 (BCD natural) o el 15 (binario natural).

En la figura 6.195*a* se representa el símbolo de un contador de flancos de bajada en binario natural de 4 biestables y en la figura 6.195*b* su cronograma de evolución en el tiempo. Se supone que el bloque posee una entrada de puesta a cero (Reset) activa con un uno lógico. Dicha entrada se debe colocar en el nivel del cero lógico para que pueda realizarse el contaje.

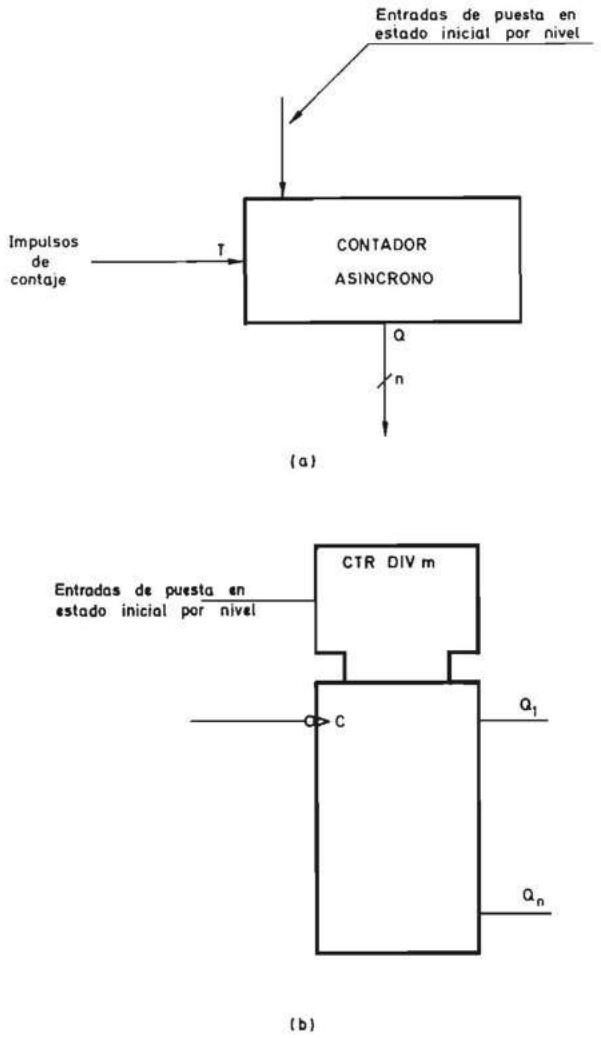
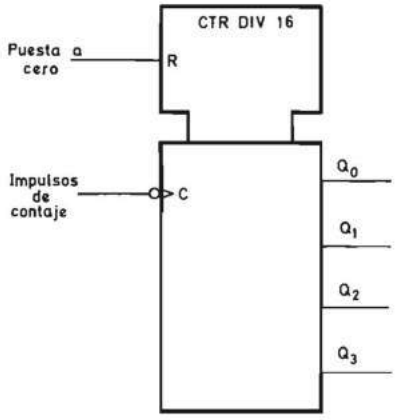


FIGURA 6.194.—**Bloque funcional contador asíncrono: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**

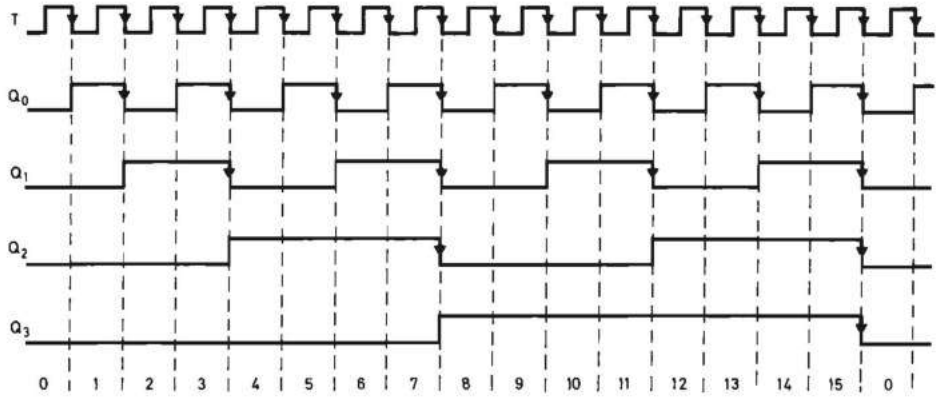
En el cronograma de la figura 6.195*b* se observa que la salida Q_3 presenta un flanco de bajada solamente cuando el contador pasa del estado 15 al 0. Esta circunstancia permite ampliar la capacidad de contaje conectando los bloques funcionales tal como se indica en la figura 6.196*a* y *b*. La salida Q_3 de cada bloque se conecta a la entrada *T(C)* del siguiente. Las entradas de puesta a cero de los diferentes bloques se conectan entre sí para obtener un terminal único que permita poner a cero todos los bloques simultáneamente.

En la figura 6.197*a* se representa el símbolo de un contador de flancos de bajada en BCD natural de 4 biestables y en la figura 6.197*b* su cronograma de evolución en el tiempo. Su capacidad de contaje máxima es 9 y, por ello, recibe el nombre de década contadora. Se supone también que el bloque posee una entrada de puesta a cero activa con un nivel uno lógico.

En el cronograma de la figura 6.197*b* se observa que la salida Q_3 presenta un flanco de bajada solamente cuando el contador pasa del estado 9 al 0. Por ello se puede ampliar la capacidad de contaje de forma idéntica a la estudiada anterior-



(a)



(b)

FIGURA 6.195.—**Contador asíncrono binario natural de cuatro biestables: *a)* Símbolo lógico normalizado; *b)* Diagrama de evolución temporal de sus salidas.**

mente para los contadores en binario natural. En la figura 6.198 se representa un contador de capacidad de 999 realizado mediante tres décadas BCD natural acopladas entre sí.

Se comprende fácilmente que la ampliación de la capacidad de la forma indicada en las figuras 6.196 y 6.198 disminuye la frecuencia máxima de contaje.

Muchas veces es necesario obtener una indicación de que un contador asíncrono ha llegado a un determinado estado. La acción de obtener una salida cuya subida al estado uno o bajada al estado cero indique que se ha alcanzado un estado se de-

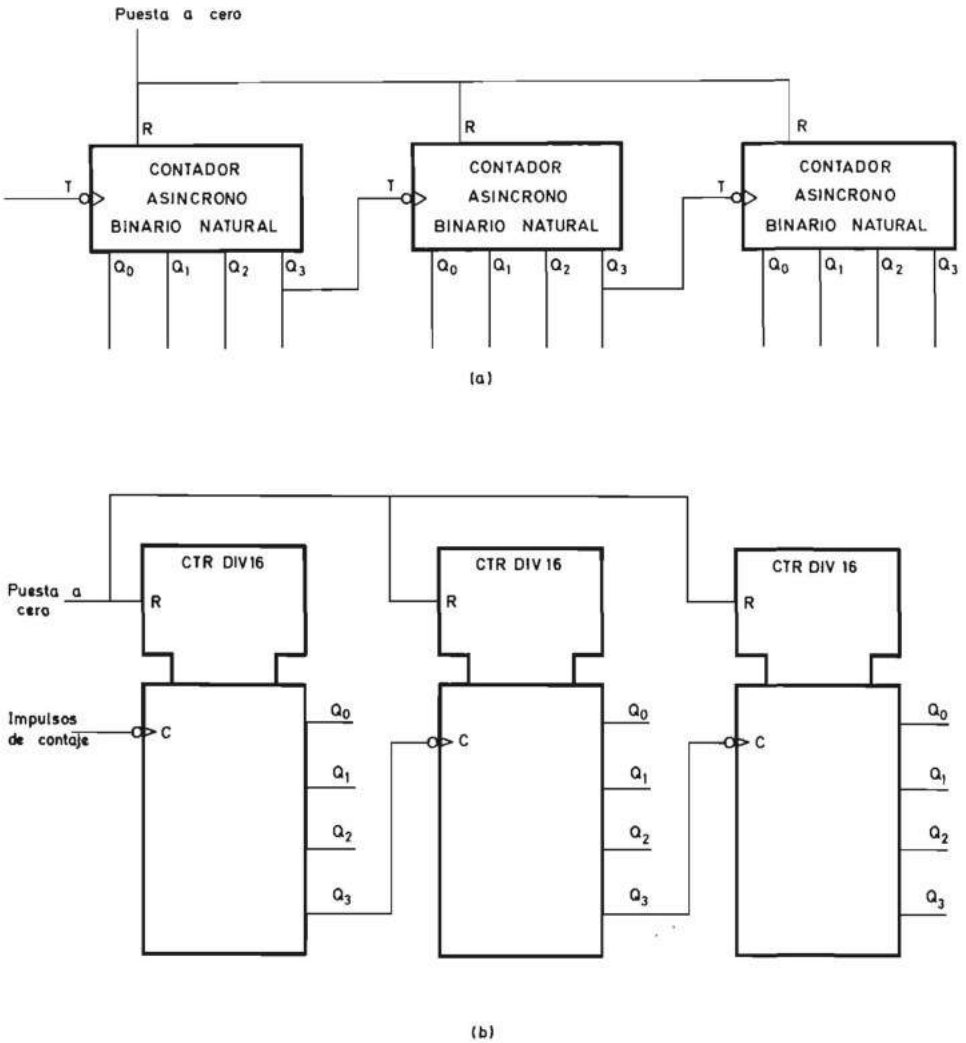


FIGURA 6.196.—**Contador asíncrono binario natural: *a*) Esquema realizado con símbolos no normalizados; *b*) Esquema realizado con símbolos normalizados.**

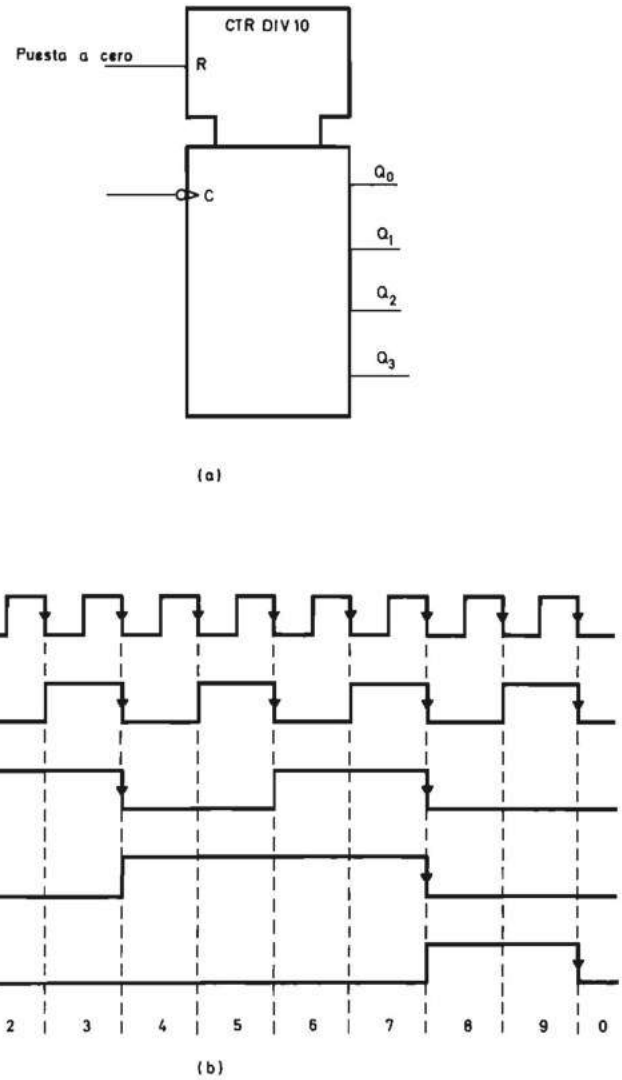


FIGURA 6.197.—**Contador asíncrono BCD natural de cuatro biestables (década binaria): *a*) Símbolo lógico normalizado; *b*) Diagrama de evolución temporal de sus salidas.**

nomina decodificar el mismo y consiste en la obtención del producto canónico correspondiente a la combinación de variables de estado interno del contador. Por ejemplo, para tener una indicación de que el contador BCD natural de la figura 6.197 ha alcanzado el estado 7, es necesario realizar el producto canónico de las variables de estado Q_0 a Q_3 que toma el valor uno cuando el contador alcanza dicho estado. Como 7 en BCD natural es equivalente a la combinación binaria

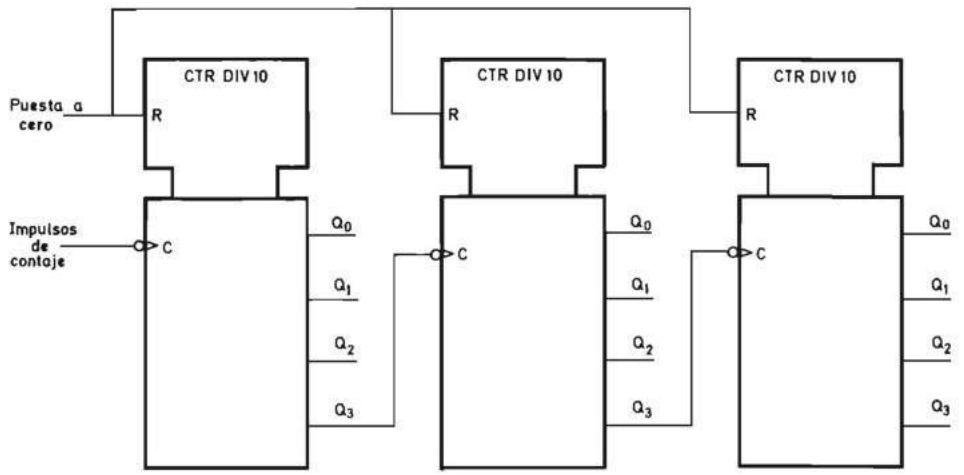


FIGURA 6.198.—**Contador asíncrono BCD natural.**

0111, el producto canónico que le corresponde es $\overline{Q}_3 Q_2 Q_1 Q_0$. En la figura 6.199 se representa la realización física mediante una puerta NO-Y que obtiene el inverso de dicho producto canónico. Dicha salida toma el estado cero solamente cuando el contador haya contado 7 impulsos.

Si el código de contaje del contador es el BCD natural o el binario natural y el número de estados a decodificar es elevado, se pueden utilizar los decodificadores integrados estudiados en el capítulo 3. En la figura 6.200 se representa una dé-

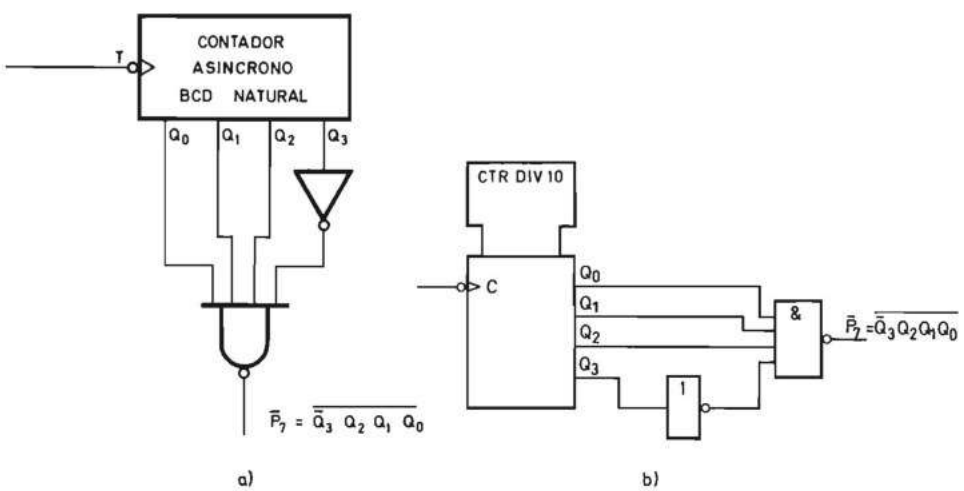
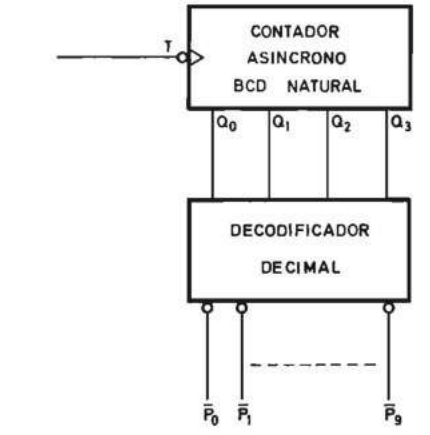


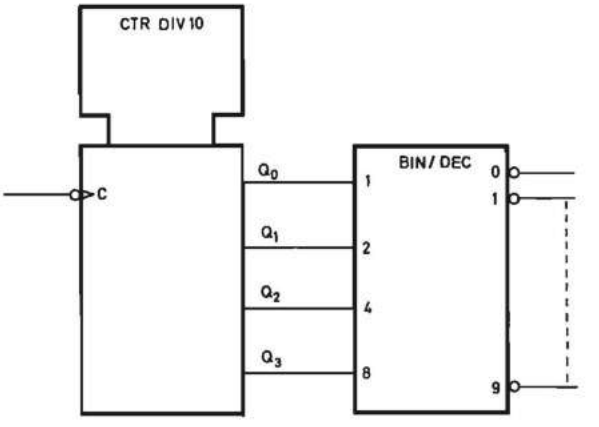
FIGURA 6.199.—**Decodificación de un contador asíncrono: *a*) Esquema realizado con símbolos no normalizados; *b*) Esquema realizado con símbolos normalizados.**

cada asíncrona BCD natural cuyos estados se decodifican mediante un decodificador decimal como los estudiados en el apartado 3.7.1.

En la decodificación de los contadores asíncronos es posible tener impulsos aleatorios debido a que los biestables no cambian de estado simultáneamente. Por ejemplo, cuando el contador en BCD natural de la figura 6.197 pasa del estado 0111 al 1000, no lo hace directamente sino que pasa por los estados intermedios 0110, 0100 y 0000. Dichos impulsos aleatorios aparecen después de los flancos activos de los impulsos de contaje y para evitarlos es necesario inhibir la decodificación durante el intervalo comprendido entre aquéllos y los flancos no activos. En



(a)



(b)

FIGURA 6.200.—**Decodificación de una década asíncrona con un decodificador decimal: *a*) Esquema realizado con símbolos no normalizados; *b*) Esquema realizado con símbolos normalizados.**

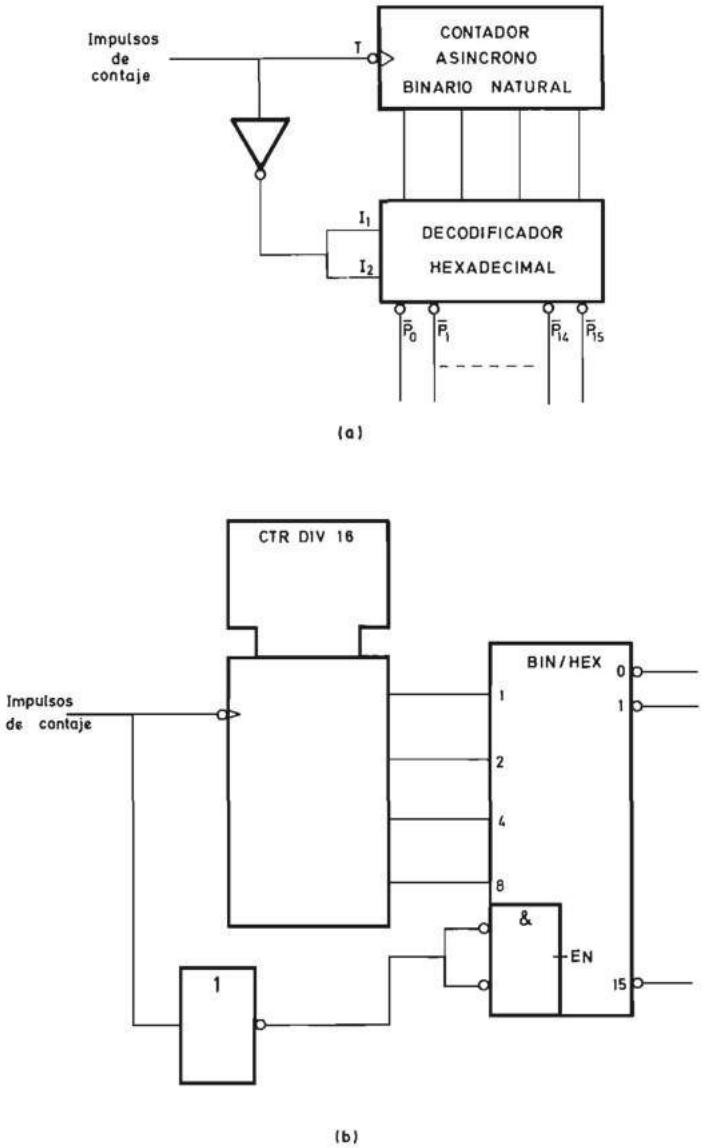


FIGURA 6.201.—**Decodificación de un contador asíncrono binario natural con un decodificador hexadecimal: *a*) Esquema realizado con símbolos no normalizados; *b*) Esquema realizado con símbolos normalizados.**

la figura 6.201 se representa la decodificación del contador asíncrono binario natural mediante el decodificador hexadecimal de la figura 3.72. La inhibición de este decodificador se realiza aplicando un uno a las entradas correspondientes y los cambios de estado del contador se producen con los flancos de bajada. Por tanto,

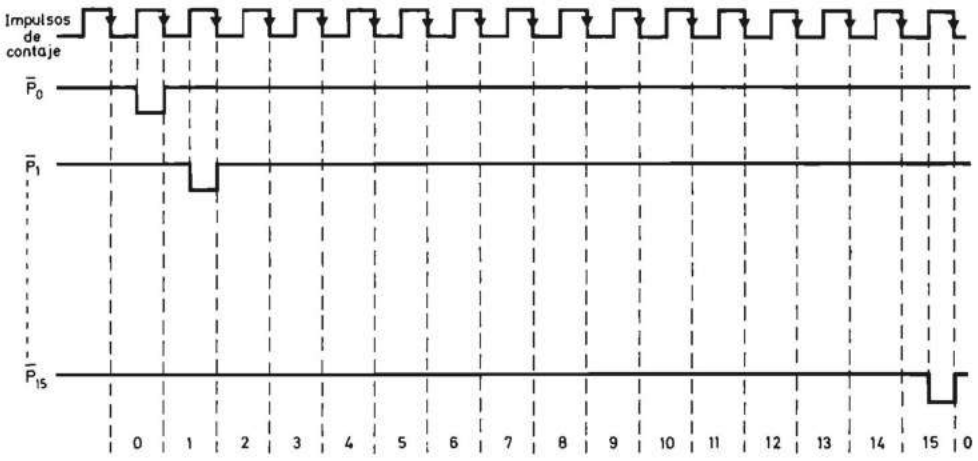


FIGURA 6.202.—**Diagrama de evolución temporal de las salidas del decodificador de la figura 6.201.**

la inhibición se realiza mediante el inverso de los impulsos de entrada. Los impulsos de salida del decodificador alcanzan el estado cero solamente cuando los impulsos de entrada se encuentran en estado uno. En la figura 6.202 se representa el diagrama de secuencia de los impulsos a la salida del decodificador.

Mediante el simple acoplamiento de bloques funcionales contadores asíncronos no es posible obtener un contador de cualquier capacidad. Cuando se desee obtener un contador de una cierta capacidad se debe proceder de la siguiente manera:

a) Acoplar el número adecuado de bloques funcionales que permita obtener un contador con una capacidad de contaje superior a la deseada y lo más cercana posible a aquélla.

b) Decodificar el estado correspondiente a la capacidad de contaje deseada y utilizar la variable binaria así obtenida para realizar la puesta a cero de los bloques funcionales.

Por ejemplo, si desea obtener un contador cuya capacidad no supere 999 se puede realizar con tres décadas (contadores en BCD natural de 4 biestables) acoplados tal como se representa en la figura 6.198.

La decodificación del estado deseado se puede realizar de diversas maneras. Una de ellas se representa en la figura 6.203. Cada década se decodifica mediante una puerta NO-Y (NAND) de cuatro entradas y un inversor. A las cuatro entradas de la puerta NO-Y (NAND) se conectan las variables directas Q o inversas $\overline{Q}$ del contador de acuerdo con el estado que se desee decodificar. Por ejemplo, si se quiere decodificar el estado 5 se conectan a las citadas entradas las variables Q_0, $\overline{Q}_1$, Q_2 y $\overline{Q}_3$. Mediante otra puerta NO-Y (NAND) N_7 de tres entradas se obtiene la decodificación del estado del contador correspondiente a la capacidad máxima. La conmutación del estado uno al cero de la salida de esta puerta indica que el número de impulsos aplicados a la entrada del contador ha alcanzado la capacidad máxima del mismo. Debido a que las décadas utilizadas son asíncronas, a la salida de

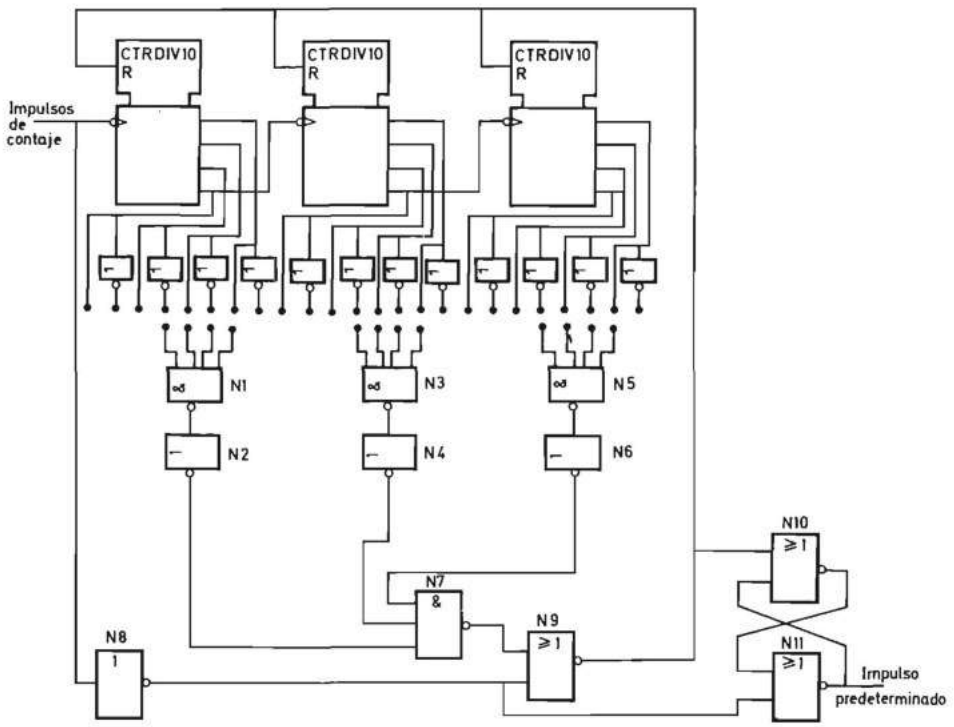


FIGURA 6.203.—**Esquema de un contador asíncrono BCD natural de capacidad variable.**

N_7 aparecen impulsos aleatorios después del flanco activo de los impulsos de entrada (en este caso particular es el de bajada). Para suprimirlos se realiza una sincronización con el inverso de los impulsos de contaje por medio de la puerta NO-O (NOR) N_9. La salida de esta puerta se pone a nivel uno cuando el contador alcanza la capacidad máxima y la entrada de impulsos vuelve al nivel uno. Si dicha salida se conecta a la entrada de puesta a cero de todas las décadas, en ese instante se ponen a cero todas ellas y el siguiente flanco de bajada de los impulsos inicia un nuevo ciclo de contaje. El impulso de puesta a cero tiene una duración igual al tiempo que tarda en realizarse esa acción sobre un biestable cualquiera de la década más el tiempo de propagación del circuito decodificador. Dicha duración ha de ser suficiente para asegurar la puesta a cero de todos los biestables de todas las décadas.

A veces se necesita tener un impulso de duración elevada para excitar algún elemento indicador (un diodo emisor de luz, por ejemplo) o provocar alguna acción externa (activación de un relé, por ejemplo). Para ello el impulso que realiza la puesta a cero puede ser memorizado en un biestable. En la figura 6.203 se utiliza un biestable *R-S* formado por las dos puertas NO-O (NOR) N_{10} y N_{11}. La puesta a cero de nuevo de este biestable se puede realizar manualmente o bien cuando se realiza el contaje del primer impulso del nuevo ciclo. Esta es la opción elegida en la figura 6.203 en la que dicha puesta a cero se realiza mediante la salida del inversor N_8.

La obtención de un contador de capacidad programable con conmutadores rotativos se puede realizar de forma similar utilizando comparadores binarios o puertas O-exclusiva. En el libro «Manual de prácticas de electrónica digital» de Editorial Marcombo, se exponen diversos montajes prácticos de contadores asíncronos con circuitos integrados de tecnología TTL y CMOS.

6.3.5.4.3.3 Contadores síncronos. Los bloques funcionales contadores síncronos responden a la teoría desarrollada en el apartado 6.3.5.2.2. La entrada de impulsos se conecta a la entrada $T(C)$ de todos los biestables. Por ser más complicado el circuito combinacional, su integración se realizó posteriormente a la de los contadores asíncronos. El interés de poder introducir información en paralelo en el interior del contador, tal como se indicará en el apartado 6.3.5.5.3.2, hizo que se integrasen bloques funcionales síncronos que responden, en general, a los símbolos no normalizado y normalizado representados en la figura 6.204*a* y *b*.

En la designación de los terminales del símbolo no normalizado no existe unanimidad por parte de los distintos fabricantes. En la figura 6.204*a* se crea una nomenclatura que sigue las líneas generales de la teoría expuesta en el apartado 6.3.5.4 y que es la siguiente.

a) La entrada T de impulsos de contaje.
b) Las entradas D de información en paralelo.
c) La entrada C/P de modo de funcionamiento que, según se encuentre en un estado lógico u otro, hace que los impulsos aplicados a T introduzcan en paralelo la información presente en D o bien que se realice el contaje en el código adecuado de acuerdo con la configuración del circuito combinacional situado en el interior del bloque y a través del cual se realimentan los biestables. Al igual que en el caso de los contadores asíncronos se han realizado bloques funcionales contadores síncronos en binario natural y BCD natural de capacidad de contaje 15 y 9 respectivamente. A la entrada C/P se la denomina $\overline{C}/P$ si se realiza el contaje cuando se encuentra en cero y la entrada en paralelo cuando se encuentra en uno y $C/\overline{P}$ si se tiene la situación contraria.
d) La entrada de inhibición I, que si se encuentra en un determinado nivel inhibe la acción de los impulsos aplicados a T. Se comprende fácilmente que existen varias posibilidades y que la inhibición puede actuar tanto sobre la entrada en paralelo como sobre el contaje o solamente sobre este último. La opción más usual es que actúa sólo sobre el contaje. Se denomina a la inhibición $\overline{I}$ si se realiza esta acción cuando se encuentra en nivel cero, e I si se realiza cuando se encuentra en nivel uno.

e) La entrada de puesta a cero R que puede ser asíncrona o síncrona. Cuando es asíncrona se realiza la acción de puesta a cero en cuanto este terminal se coloca en el nivel activo que puede ser el uno (se la denomina R) o cero (se la denomina $\overline{R}$). Cuando es síncrona se realiza la acción de puesta a cero cuando este terminal se encuentra en el nivel activo y, además, se aplica un flanco activo a la entrada de impulsos T.

En la figura 6.204*b* se representa el símbolo normalizado que posee los siguientes terminales:

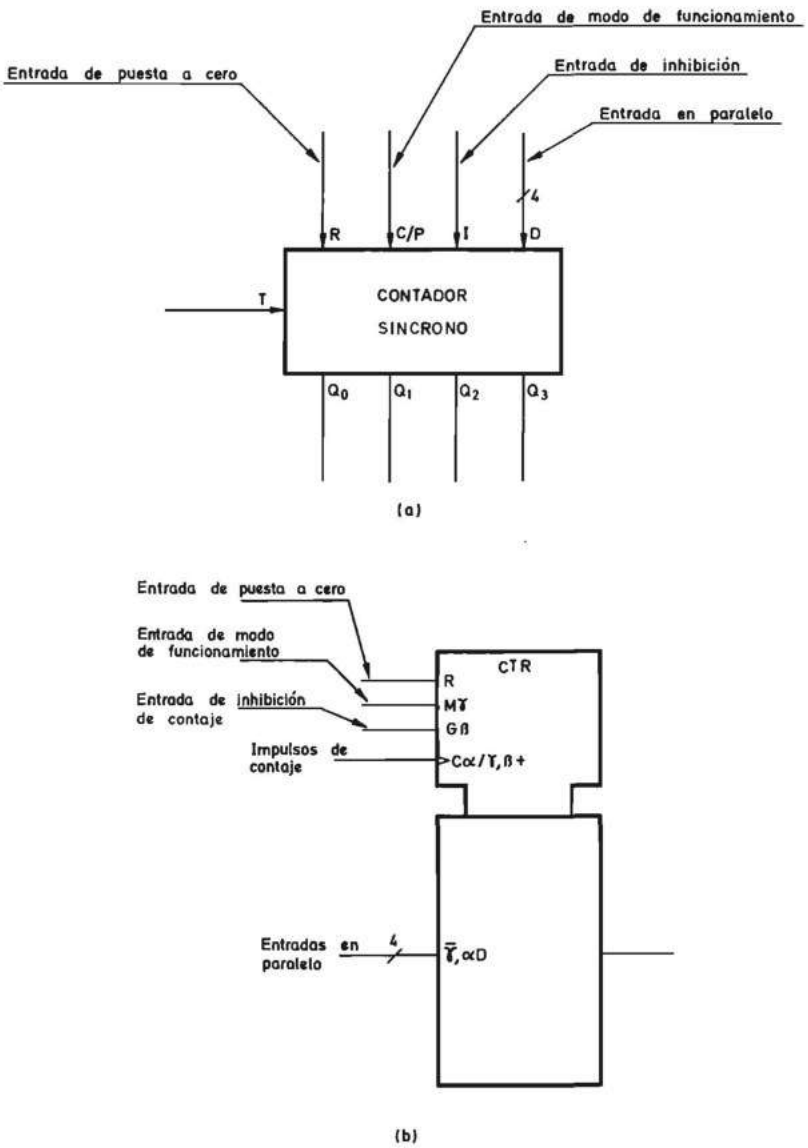


FIGURA 6.204.—**Bloque funcional contador síncrono: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**

R : Entrada puesta a cero asíncrona activa con nivel uno.

$M\gamma$: Entrada de modo de operación. Si se encuentra el nivel cero, la información entra en paralelo, y si se encuentra en nivel uno, se realiza el contaje.

$G\beta$: Entrada de inhibición de contaje, que permite que se realice cuando se encuentra en nivel uno.

$C\alpha/\gamma,\beta+$: Entrada de impulsos de sincronismo que provoca la entrada de información en paralelo ($C\alpha$) o el contaje en sentido ascendente (+) si $M\gamma = G\beta = 1$

$\bar{\gamma},\alpha D$: Entrada de información en paralelo. Se realiza esta acción al aplicar un flanco activo en $C\alpha$ (el de subida en este caso) si $M\gamma = 0$.

Para ampliar la capacidad de contaje es posible acoplar los bloques funcionales síncronos tal como se hizo con los asíncronos en las figuras 6.196 y 6.198. Pero esta forma de ampliar la capacidad de contaje hace que el contador así obtenido deje de ser síncrono con la consiguiente disminución de la frecuencia máxima de contaje y la aparición de impulsos aleatorios en la decodificación.

Por ello para que se pueda realizar la ampliación conectando los impulsos de contaje a la entrada T de todos los bloques, es necesario que éstos posean una entrada de propagación E_p y una salida de propagación S_p indicadas en la figura 6.205*a* que representa el símbolo no normalizado.

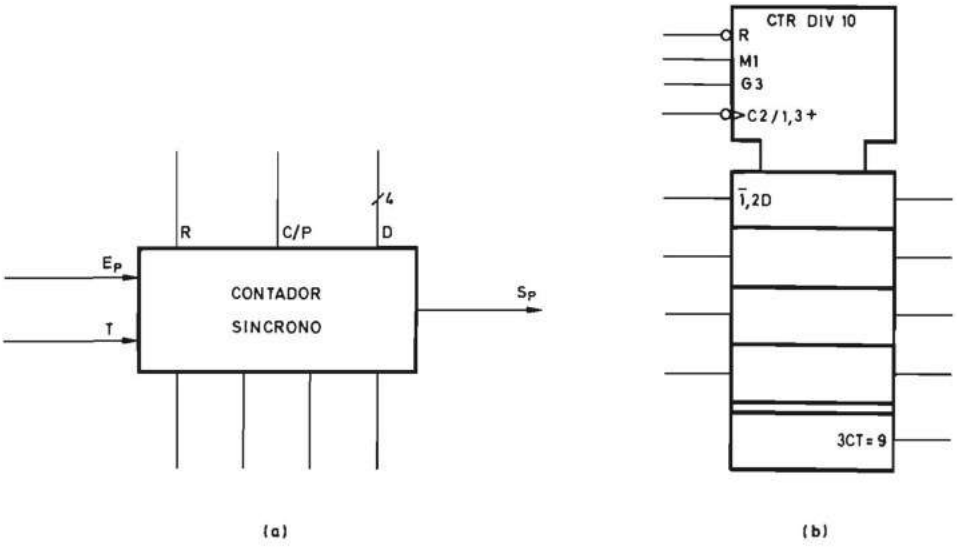


FIGURA 6.205.—**Símbolo lógico normalizado de un contador síncrono BCD natural: *a*) No normalizado; *b*) Normalizado.**

La entrada E_p debe permitir el contaje cuando se encuentra en un cierto nivel que se puede ser el cero o un lógico o inhibirlo en caso contrario. Se la denomina E_p si permite el contaje cuando se encuentra en nivel uno. Comparando la forma de actuación de E_p con la de la entrada de inhibición se deduce que son idénticas cuando esta última inhibe solamente el contaje. Por ello, en el bloque de la figura 6.205*a* no se indica el terminal I por coincidir con E_p. Pero el contaje de un bloque debe desinhibirse solamente cuando el bloque anterior se encuentre en el último estado de su capacidad máxima.

Por ello, si el bloque cuenta en BCD natural, se genera la salida de propagación S_p de tal manera que sea igual al siguiente producto lógico:

$$S_p = E_p \; Q_0 \; \overline{Q}_1 \; \overline{Q}_2 \; Q_3$$

que equivale al producto de E_p por el estado 9 del contador.

En la figura 6.205*b* se representa el símbolo normalizado que es idéntico al de la figura 6.204 con la excepción de la parte inferior en la que se ha añadido la salida de propagación S_p que lleva la indicación $3CT=9$.

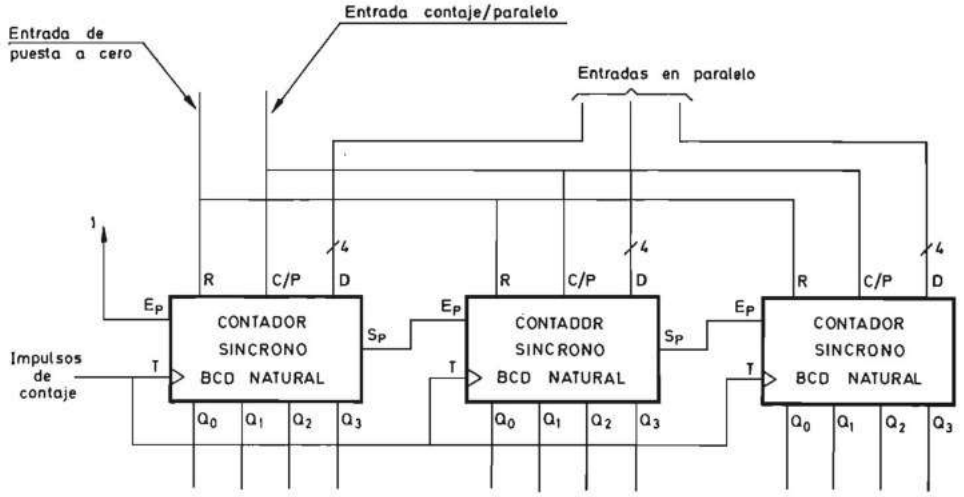
El número 3 indica que para que ese terminal se ponga a nivel uno es necesario que *G*3 se encuentre en dicho estado y $CT=9$ especifica que el contenido del contador (*CT*) ha de ser 9 para que se produzca la citada circunstancia. La colocación del indicativo 3 y el $CT=9$ correlativos indica una relación de simultaneidad.

En la figura 6.206*a* y *b* se representa un contador de capacidad máxima de contaje 999 realizado mediante el acoplamiento de tres contadores síncronos en BCD natural de capacidad 9 cada uno de ellos. Las entradas *R* de todos los bloques se unen entre sí para obtener un único terminal de puesta a cero. Lo mismo se hace con el terminal de modo de funcionamiento contaje/paralelo *C/P* (*M*1). Las entradas de los impulsos de contaje *T* (*C*2/1,3+) de todos los bloques se unen entre sí y al terminal único se aplican los citados impulsos. La entrada E_p (*G*3) del primer bloque se conecta al nivel lógico uno para que su contaje no esté nunca inhibido. Por el contrario, la E_p (*G*3) del segundo bloque se conecta a la salida S_p ($3CT=9$) del primero. De esta forma, sólo cuando el primer bloque se encuentra en el estado 9, la señal S_p ($3CT=9$) se pone en nivel uno y el siguiente impulso de contaje hace que se incremente en una unidad el contenido del segundo bloque. La entrada E_p (*G*3) del tercer bloque se conecta a la salida S_p (3CT = 9) del segundo. Se comprende fácilmente que el contenido del tercer bloque se incrementa en una unidad, al aplicar un flanco activo de los impulsos de contaje, solamente cuando los dos primeros bloques se encuentran ambos en el estado 9.

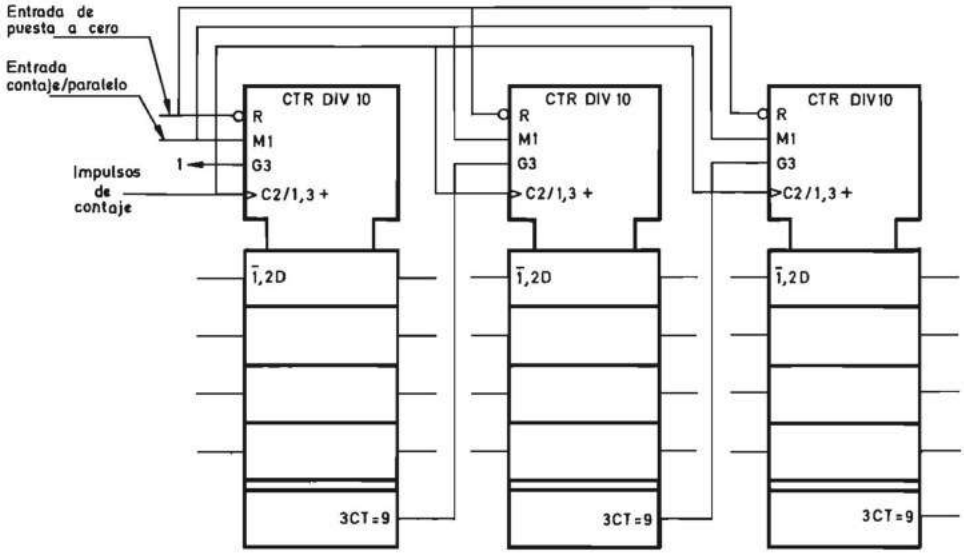
El montaje de la figura 6.206 constituye un contador síncrono porque los impulsos de contaje se aplican a las entradas *T* de todos los bloques funcionales y además presenta la ventaja de necesitar muy pocas interconexiones entre los bloques. Por el contrario, presenta el inconveniente de que la frecuencia máxima de contaje del conjunto es menor que la de un solo bloque. Ello es debido a que la llegada al último estado del primer bloque se propaga en serie a través de los demás para llegar al último. Por ejemplo, si los dos primeros bloques de la figura 6.206 se encuentran en el estado 98 y se aplica un nuevo impulso, el paso de 8 a 9 del primero, no se detecta en la salida S_p ($3CT=9$) del segundo hasta que dicho cambio se propaga a través de ambos bloques. Para superar ese inconveniente se puede dotar al bloque funcional de una entrada adicional de inhibición de contaje que inhibe el contaje sin influir en la salida S_p del propio bloque. En la figura 6.207*a* esta entrada se denomina I_c y en la figura 6.207*b*, *G*4. Para que se realice el contaje es necesario que E_p (*G*3) e I_c (*G*4) se encuentren ambas en un nivel lógico determinado (en este caso el nivel uno).

En la figura 6.208*a* y *b* se representa el esquema de un contador síncrono realizado con el bloque de la figura 6.207*a* y *b*. Las entradas E_p (*G*3) e I_c (*G*4) del

primer bloque se conectan a nivel uno. La entrada I_c ($G4$) de los demás bloques se conecta a la salida S_p ($3CT=9$) del primero; de esta forma es necesario que el contenido del primer contador sea 9 para que los demás incrementen su contenido al recibir un impulso de contaje. Además, la entrada E_p ($G3$) del segundo bloque



(a)



(b)

FIGURA 6.206.— **Esquema de un contador síncrono con entrada y salida de propagación realizado: *a*) Con símbolos lógicos no normalizados; *b*) Con símbolos normalizados.**

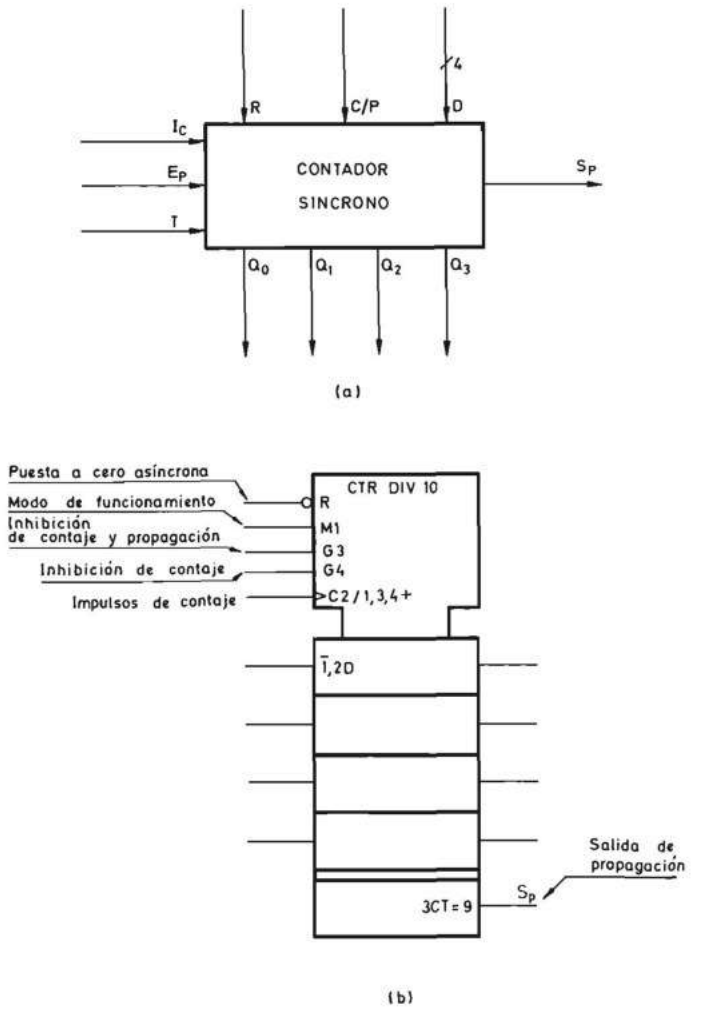
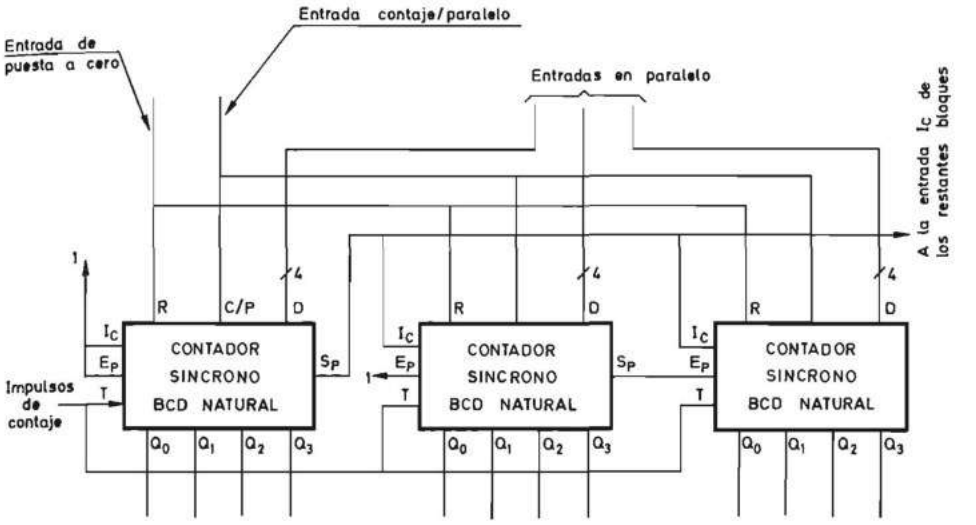


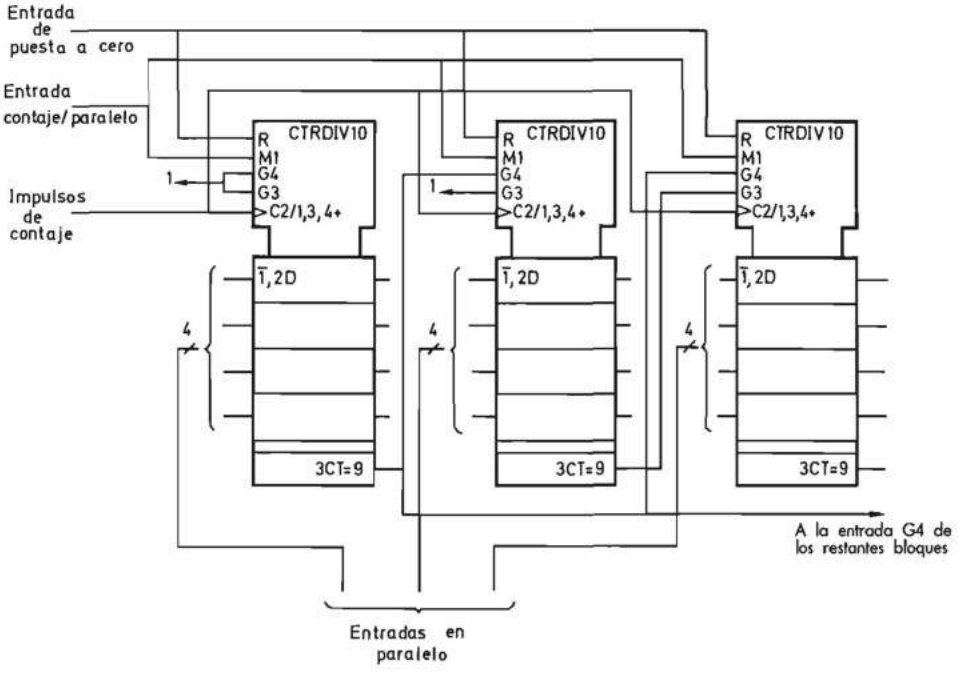
FIGURA 6.207.—**Bloque funcional contador síncrono: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado.**

se conecta a nivel uno y la entrada E_p ($G3$) de cada uno de los demás se conecta a la salida S_p ($3CT=9$) del anterior. Se logra así que para que un bloque incremente su contenido, deban estar en estado 9 todos los anteriores. Mediante este montaje se logra que la llegada al último estado (en este caso el 9) del primer bloque se trasmita de inmediato a los demás y que la frecuencia máxima de contaje del conjunto sea igual a la de un solo bloque.

De una forma similar a la estudiada para los contadores asíncronos es posible realizar un contador síncrono de cualquier capacidad de contaje. Para ello es interesante resaltar que en los contadores síncronos no existe la posibilidad de apari-



(a)



b)

FIGURA 6.208.—**Esquema de un contador síncrono BCD natural realizado: *a*) Con símbolos lógicos no normalizados; *b*) Con símbolos lógicos normalizados.**

ción de impulsos aleatorios al decodificar un estado porque todos los biestables cambian simultáneamente y, por lo tanto, no es necesario inhibir la decodificación con los impulsos de entrada.

En la figura 6.209 se representa una posible solución. La salida S_p ($3CT=9$) del último bloque se conecta invertida a la entrada $M1$ de todos ellos. De esta forma, cuando el contador alcanza la máxima capacidad, $M1$ pasa al nivel cero y el próximo impulso aplicado a la entrada T ($C2/1,3+$) introduce en paralelo la información presente en las entradas D ($\bar{1},2D$). Si N es la capacidad máxima del contador y M es la información numérica aplicada a las entradas D, la capacidad de contaje del sistema es N-M. En la figura 6.209 se representa el esquema de un contador de capacidad 654 realizado mediante tres bloques funcionales contadores síncronos en BCD natural como el representado en la figura 6.205.

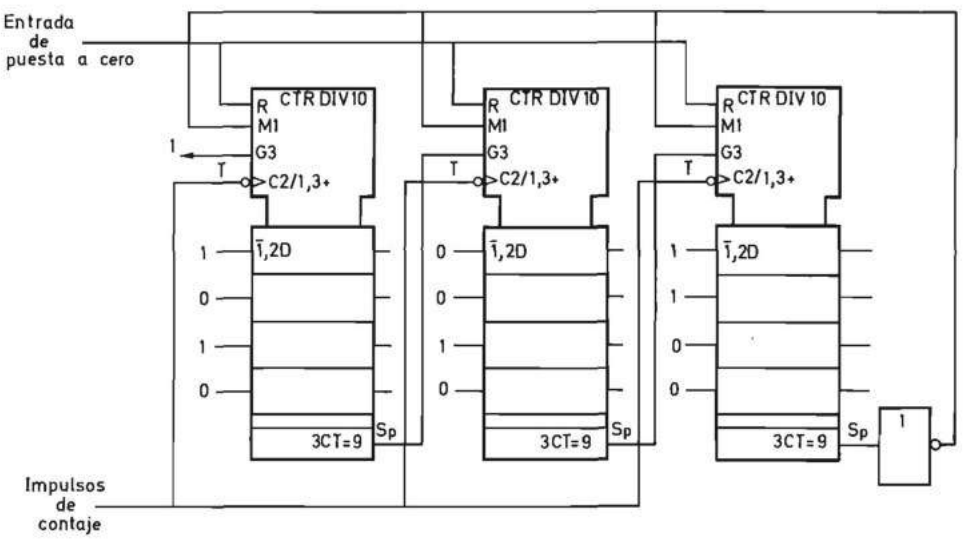


FIGURA 6.209.—**Esquema de un contador síncrono BCD natural de capacidad variable realizado con símbolos lógicos normalizados.**

El número aplicado a las entradas D es el equivalente a 345_{10} (999 — 654 = = 345), es decir, 0011 0100 0101 en BCD natural. Un inconveniente de este contador es que su contenido no es equivalente al número de impulsos contados. Se logra esto último si se decodifica el estado correspondiente a la máxima capacidad de contaje y con la salida del decodificador se controla $M1$. En la figura 6.210 se representa el esquema simplificado de esta solución. El decodificador (X/Y) puede realizarse de forma idéntica a la del contador asíncrono de la figura 6.203. Las entradas D ($\bar{1},2D$) de todos los bloques se unen al nivel lógico cero y $C/\bar{P}$ ($M1$) se pone a nivel cero cuando el contador alcanza su último estado. De esta forma el siguiente impulso pone a cero el contador. La ventaja de conectar la salida del de-

codificador a $C/\bar{P}$ ($M1$) en lugar de a la puesta a cero asíncrona R (tal como se hizo con los contadores asíncronos en la figura 6.203) es la de que la salida del decodificador permanece activada un tiempo igual al que transcurre entre dos impulsos de contaje y no es necesario el biestable formado por N_{10} y N_{11} en la figura 6.203. Es necesario, sin embargo, tener en cuenta que en el montaje de la figura 6.210 que se ha analizado se utiliza un impulso de contaje para pasar del último estado al cero.

Los bloques funcionales que se acaban de estudiar son contadores síncronos con entradas en paralelo síncronas y entradas de inhibición que realizan el contaje en BCD natural o en binario natural. El aumento de la capacidad de integración ha permitido la realización de contadores síncronos que, además de las funciones anteriores, poseen la de poder realizar el contaje en sentido ascendente o descendente según el estado de una variable binaria denominada A/D. Es posible también la realización de un contador que efectúe el contaje en binario natural o en BCD natural según el estado de otra variable binaria que se puede denominar B/D).

El lector comprenderá fácilmente que, al ser los contadores bloques funcionales relativamente complejos, con un número elevado de conceptos que presentan diversas alternativas, el fabricante de circuitos integrados puede adoptar soluciones diferentes al integrarlos.

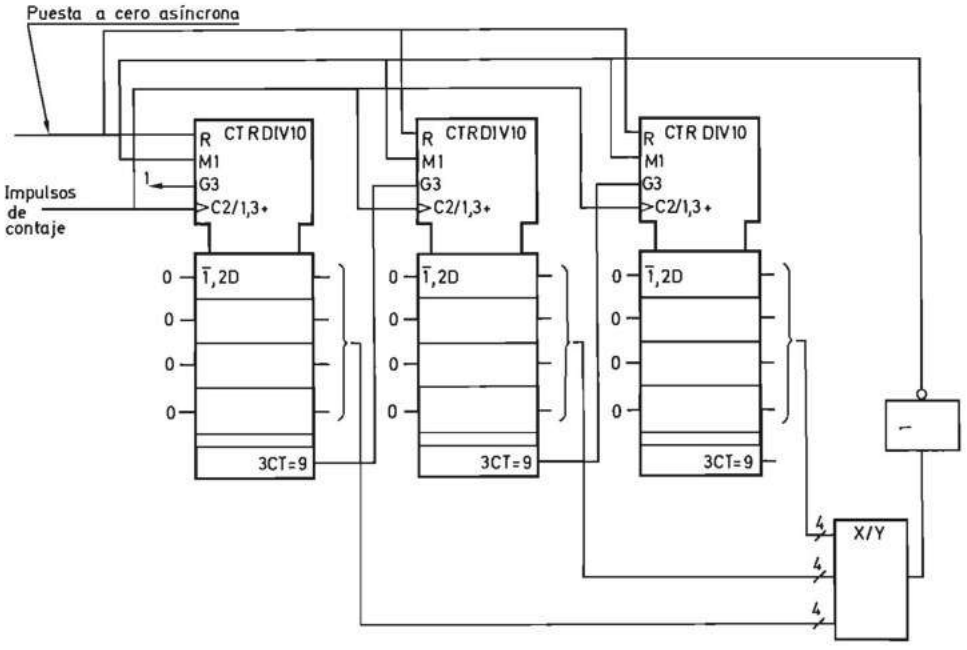


FIGURA 6.210.—**Esquema de un contador síncrono BCD natural de capacidad variable que utiliza un decodificador y está realizado con símbolos lógicos normalizados.**

Entre el apartado 6.3.5.2.2. y éste se ha procurado dar al lector los diferentes conceptos básicos que, una vez comprendidos, le permitirán analizar sin dificultad las hojas de características de los bloques funcionales integrados suministrados por los fabricantes en las diferentes tecnologías.

En el «Manual de prácticas de electrónica digital» (3.ª edición), se proponen también diversos ejercicios de contadores síncronos con circuitos integrados de las tecnologías TTL y CMOS.

6.3.5.4.3.4 Aplicaciones de los contadores. Los contadores son uno de los bloques funcionales más universales y, por ello, se pueden utilizar en la realización de la práctica totalidad de los sistemas digitales.

Por ello en este apartado se estudia solamente su aplicación directa al contaje y el resto de aplicaciones se analizan en sucesivos apartados:

a) En el apartado 6.3.5.5 se estudia la realización de sistemas secuenciales síncronos de control con contadores.
b) En el apartado 6.3.5.6.2 combinado con el tomo II se estudian las aplicaciones de los contadores a la síntesis de procesadores digitales secuenciales.
c) El apartado 6.3.5.6.3 se dedica a los controladores lógicos síncronos y su realización con contadores síncronos.

6.3.5.4.3.4.1 Sistemas contadores y su visualización. Como se indicó anteriormente, la aplicación inmediata de los contadores es el contaje de impulsos. En este tipo de aplicación resulta imprescindible la visualización (display) del contenido del contador mediante los elementos adecuados basados en fenómenos optoelectrónicos. Los de uso más extendido son los basados en la emisión de luz por parte de ciertos diodos a través de los que se hace pasar una corriente directa de un cierto valor.

Para facilitar la conversión del binario al decimal se utilizan décadas contadoras en BCD natural.

La visualización del estado en que se encuentra en cada instante una década se puede realizar mediante un visualizador de 7 segmentos constituido por diodos emisores de luz que se representa en las figuras 6.211 y 6.212. La forma del visualizador es la de 7 barras que se iluminan adecuadamente para representar los números en un formato o código de siete segmentos. A cada segmento está asociado un diodo emisor de luz (LED) que emite luz cuando se hace pasar por él una corriente adecuada (alrededor de 10 mA en los visualizadores más usuales). Desde un punto de vista eléctrico los diodos de cada barra tienen un terminal común que se une a uno de los polos de una fuente de alimentación. En la figura 6.211 el terminal común es el ánodo y el visualizador se llama de ánodo común. Los cátodos de los diodos se unen a un terminal exterior del visualizador y éste se conecta a una resistencia R. Si el otro extremo de una de las resistencias se une al otro terminal de la fuente de alimentación, el diodo correspondiente se iluminará. De acuerdo con el número que se requiere visualizar es necesario conectar a dicho terminal las resistencias adecuadas.

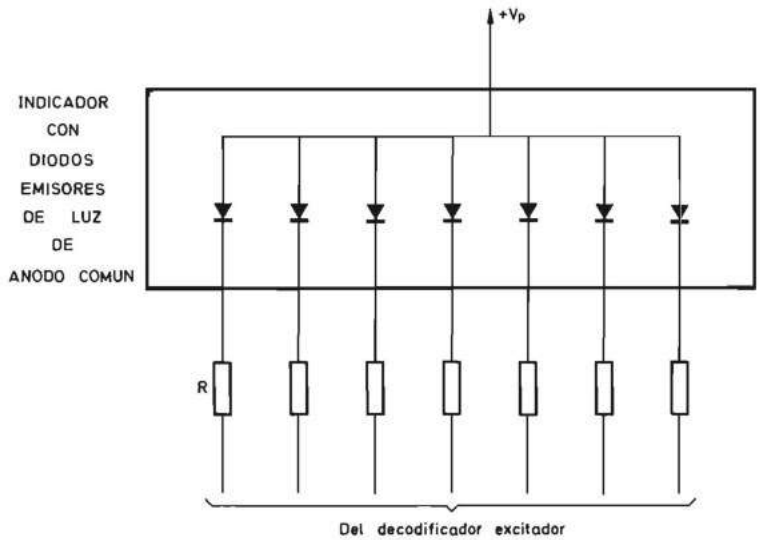


FIGURA 6.211.—**Esquema de un visualizador con diodos emisores de luz (LED) de ánodo común.**

En la figura 6.212 se representa el diagrama de bloques del visualizador.

A continuación se realiza la visualización del contador de la figura 6.208. Para visualizar el contenido de cada década con un visualizador como los que se acaban de describir se necesita colocar entre ambos un circuito que realice las dos funciones siguientes:

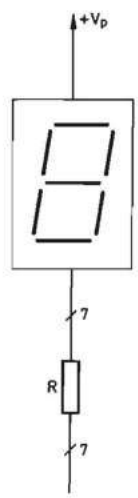


FIGURA 6.212.—**Diagrama de bloques de un visualizador con diodos emisores de luz (LED).**

a) Convertidor del código BCD natural al de siete segmentos del visualizador.

b) Interruptores de salida controlados por el convertidor que permitan el paso a través de ellos de la corriente de los diodos.

Este circuito ha sido realizado en escala de integración media en diversas tecnologías bajo el nombre de decodificador excitador (decoder driver). En la figura 6.213 se representa el diagrama de bloques de un decodificador excitador en el que se supone que los interruptores de salida son transistores bipolares. Según cuál sea el número presente a las entradas, el convertidor envía corriente a las bases de los transistores de salida adecuados llevándolos al estado de saturación que permite el paso de corriente a través de la resistencia R y el diodo adecuado. Los demás transistores permanecen cortados, es decir, no permiten el paso de la corriente a su través.

En la figura 6.214 se representa el esquema total. La salida de cada década se une a un decodificador y éste a su vez a un visualizador. Para simplificar el dibujo se representan con una sola línea las entradas y salidas del decodificador, atrave-

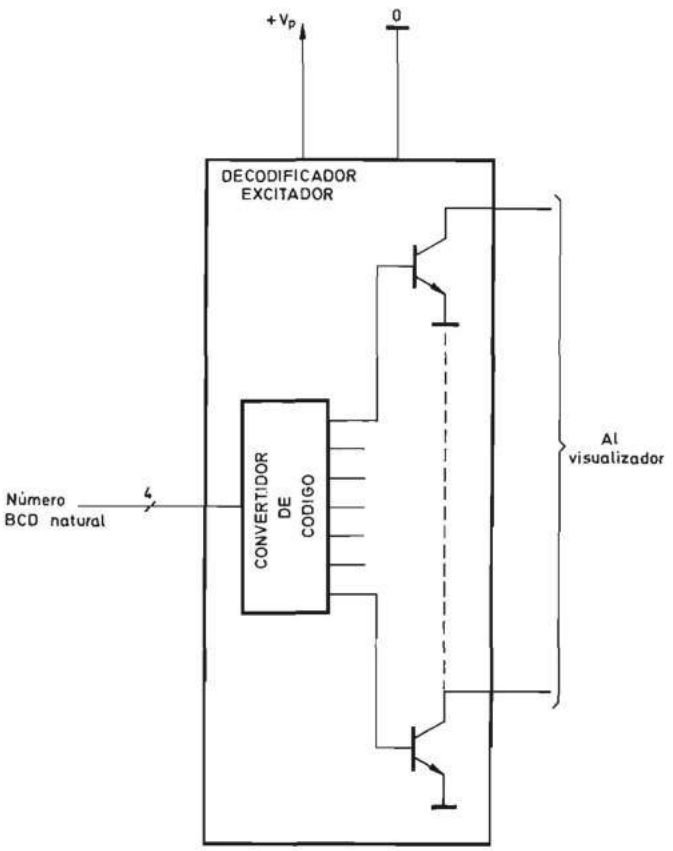


FIGURA 6.213.—**Circuito decodificador excitador (decoder driver).**

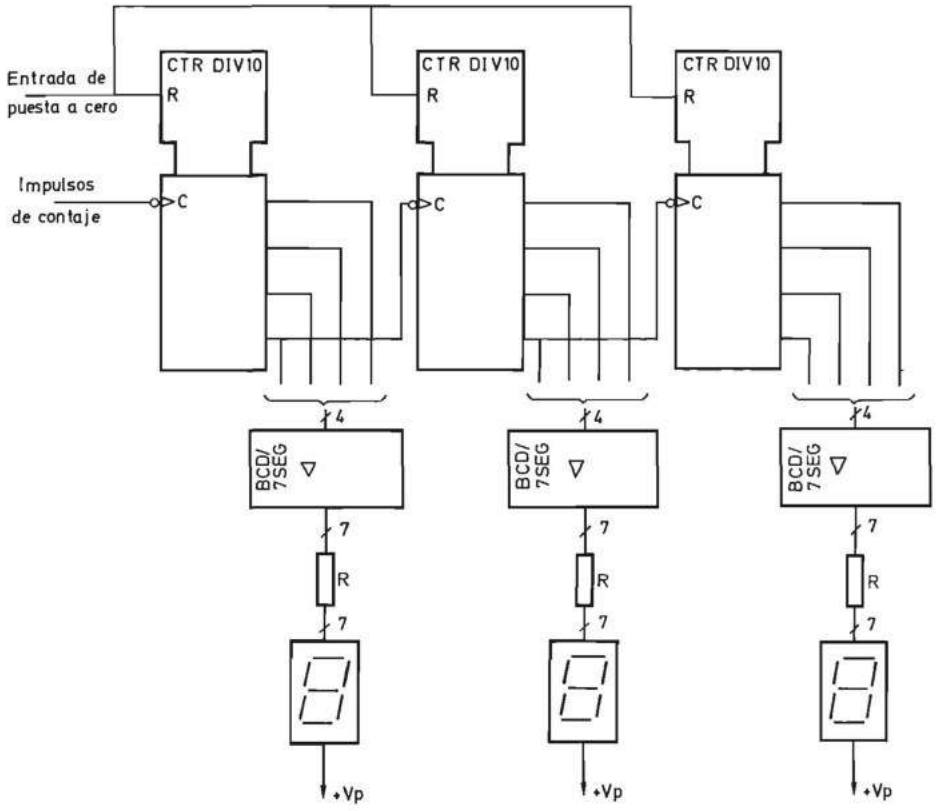


FIGURA 6.214.—**Esquema de un contador asíncrono BCD natural con visualización estática.**

sándola por un segmento a cuyo lado se indica el número de conexiones. El valor de R es aproximadamente igual a:

$$R \approx \frac{V_p - 1\ \text{V}}{10\ \text{mA}}\ \text{k}\Omega$$

donde: V_p = tensión de alimentación en voltios
1 V = caída de tensión aproximada en el diodo emisor de luz y en el transistor conmutador.

El visualizador de la figura 6.214, que utiliza un decodificador excitador para cada una de las décadas, recibe el nombre de visualizador estático porque la corriente pasa de forma continua a través del visualizador.

El progreso de las técnicas de integración ha permitido la realización de bloques integrados de elevada complejidad, especialmente en tecnologías de transistores unipolares (MOS).

Aunque los distintos bloques funcionales están diseñados de acuerdo con una teoría general, al elevarse su complejidad aparecen numerosas alternativas que los distintos fabricantes resuelven de forma diferente.

El usuario de esos bloques funcionales debe conocer la teoría general para pasar después a estudiar las características específicas de cada bloque funcional y seleccionar aquel que se adapte mejor a las necesidades de la aplicación concreta.

Las características generales más importantes de un bloque funcional contador son:

a) Capacidad de contaje. En especial en aplicaciones industriales se utiliza el código decimal codificado en binario natural (BCD natural) y por ello la capacidad del contador se mide en dígitos. Por ejemplo un contador de tres dígitos permite realizar el contaje desde cero hasta 999 en BCD natural.

b) La frecuencia máxima de los impulsos de contaje. En especial en aplicaciones industriales las frecuencias de contaje son bajas y los contadores de las distintas tecnologías admiten impulsos de frecuencia superior a la mínima necesaria.

c) Posibilidad de visualización del contenido del contador mediante un indicador numérico con diodos emisores de luz (LED) o cristales líquidos. Esta es una característica imprescindible en todo contador.

d) Disponibilidad de entradas de puesta en estado inicial que permitan colocar el contador en un determinado estado a partir del cual se inicie el contaje.

e) Salida de propagación que permita el acoplamiento de bloques funcionales idénticos para obtener un contador de mayor capacidad.

La problemática que aparece al integrar un contador de capacidad elevada es el gran número de terminales que se necesitan para realizar la visualización de la forma estática representada en la figura 6.214. La reducción de terminales se logra realizando la visualización de forma dinámica. En la figura 6.215 se representa un contador asíncrono de capacidad 9999 unido a un visualizador dinámico. En este esquema se pueden analizar los conceptos básicos de los contadores visualizados dinámicamente que han sido realizados como bloques funcionales de gran escala de integración (LSI).

Los visualizadores están unidos entre sí y a un único decodificador excitador que puede recibir en sus entradas la información de cualquiera de las décadas contadoras seleccionadas a través de un circuito multiplexor.

El otro extremo de cada visualizador se conecta a un interruptor controlado, que a su vez está unido a un terminal de la fuente de alimentación. Simultáneamente se ha de activar un interruptor y presentar en las entradas del decodificador excitador la información de la década correspondiente. De forma secuencial y periódica se han de ir activando los interruptores. Esto se logra haciendo que las entradas de selección del multiplexor se gobiernen por medio de un contador a cuya entrada T se aplican los impulsos de un generador y controlando los interruptores mediante un decodificador unido a las salidas del contador (fig. 6.215). Eligiendo adecuadamente la frecuencia del generador, a un observador humano le parecerá que todos los dígitos están activados simultáneamente.

En la figura 6.215 se representó un contador de cuatro décadas con visualización dinámica. En el mismo se analizaron las funciones imprescindibles en todo bloque funcional contador, pero existen muchas opciones que dan gran flexibilidad al contador, por ejemplo:

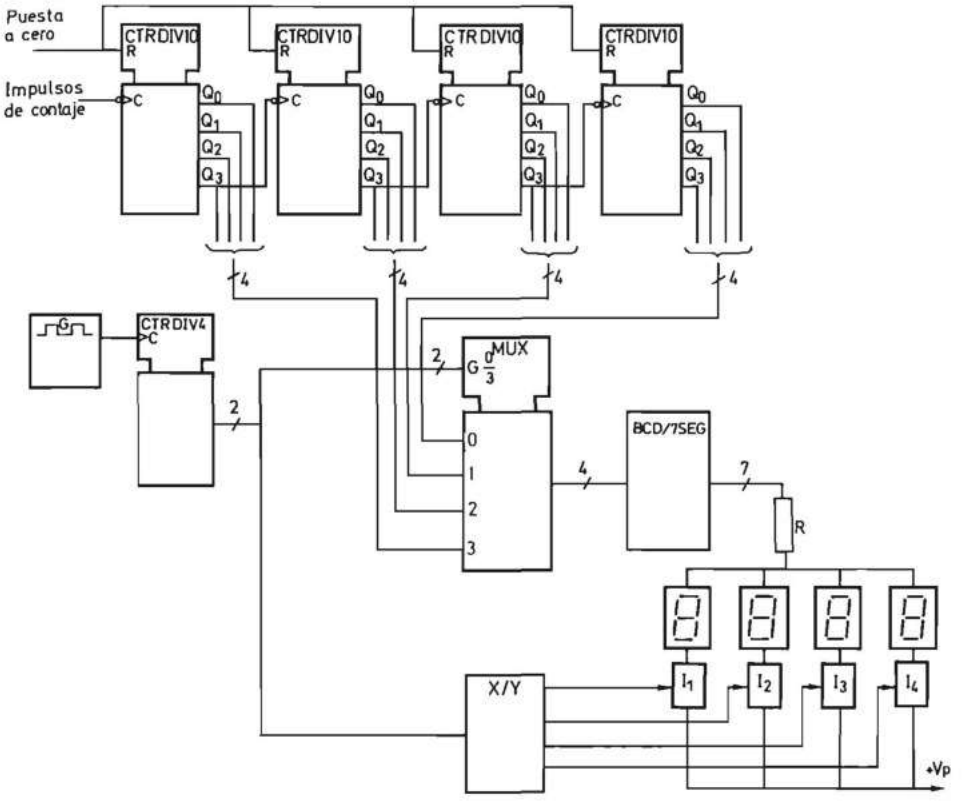


FIGURA 6.215.—**Esquema de un contador asíncrono BCD natural con visualización dinámica.**

a) La colocación de un registro temporal a la salida del contador para transferir el contenido de éste y visualizarlo mientras se realiza un nuevo ciclo de contaje.

b) Un circuito de inhibición de visualización cuya misión sea el que no se visualicen los ceros no significativos.

c) Que el contador sea reversible y permita la realización del contaje en sentido ascendente o descendente según el estado de una variable binaria.

d) Que la capacidad de contaje sea programable, para lo cual se incluyen en el interior del bloque funcional los circuitos comparadores estudiados en el capítulo 3.

e) Que además de ser visualizada, la información contenida en el contador pueda ser transferida en forma de señales eléctricas a otro sistema electrónico como, por ejemplo, un microcomputador.

El número tan elevado de opciones ha hecho que varios fabricantes de semiconductores hayan desarrollado bloques funcionales contadores diferentes en circuitos integrados de gran escala de integración que poseen todos en común la característica de contener los circuitos adecuados para acoplarlos a un visualizador dinámico. Para un estudio detenido se remite al lector a la bibliografía [INTER 81*b*] [MAND 95].

6.3.5.4.3.4.2 Divisores de frecuencia. Un divisor de frecuencia es un circuito que, aplicándole a su entrada impulsos de una frecuencia determinada, proporciona a su salida un número de impulsos por segundo que es una fracción de los que se aplican a su entrada.

Todo contador puede ser utilizado como divisor. La capacidad de división es igual a la capacidad de contaje más uno. Si se decodifica un estado cualquiera de un contador cuya capacidad de contaje es n, en la salida se obtendrá un impulso por cada $n + 1$ aplicados a su entrada.

Por ejemplo, el contador hasta 9 de la figura 6.197, es un divisor por diez, porque cada diez impulsos de entrada, el sistema vuelve al mismo estado interno.

Un divisor se puede representar mediante un diagrama de bloques como el indicado en la figura 6.216.

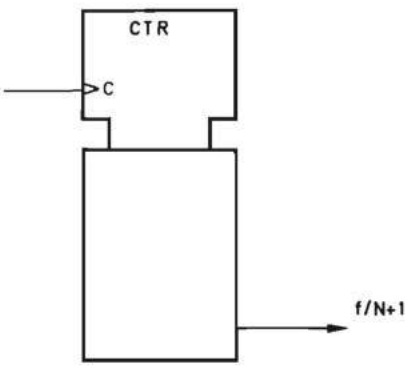


FIGURA 6.216.—**Divisor de frecuencia.**

El circuito de la figura 6.209, del que se indicaron los inconvenientes como contador, constituye un excelente divisor que sólo utiliza bloques funcionales contadores. Modificando el valor del número binario colocado en las entradas D de los bloques se puede obtener un divisor cuya capacidad de división es programable.

6.3.5.4.4 Registros de desplazamiento

6.3.5.4.4.1 Generalidades. Como se vio en el apartado 6.3.5.2.3 los registros de desplazamiento son sistemas secuenciales síncronos en los que los biestables se conectan de tal manera que cuando se aplica un flanco activo a la entrada T común a todos ellos, la información presente en la entrada se introduce en el primero, la de éste pasa al segundo, la del segundo al tercero, y así sucesivamente.

En la figura 6.217 se representa el esquema de un registro de desplazamiento compuesto por biestables D activados por flancos de subida. Se supone que los biestables carecen de puesta a uno asíncrona y poseen una puesta a cero asíncrona

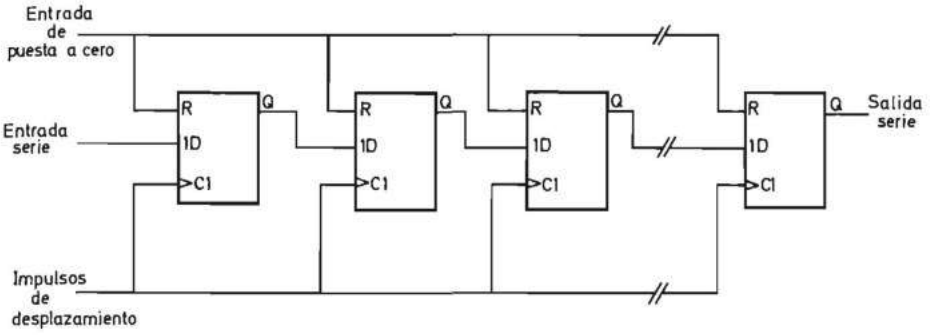


FIGURA 6.217.—**Esquema de un registro de desplazamiento realizado con biestables *D* activados por flancos de subida.**

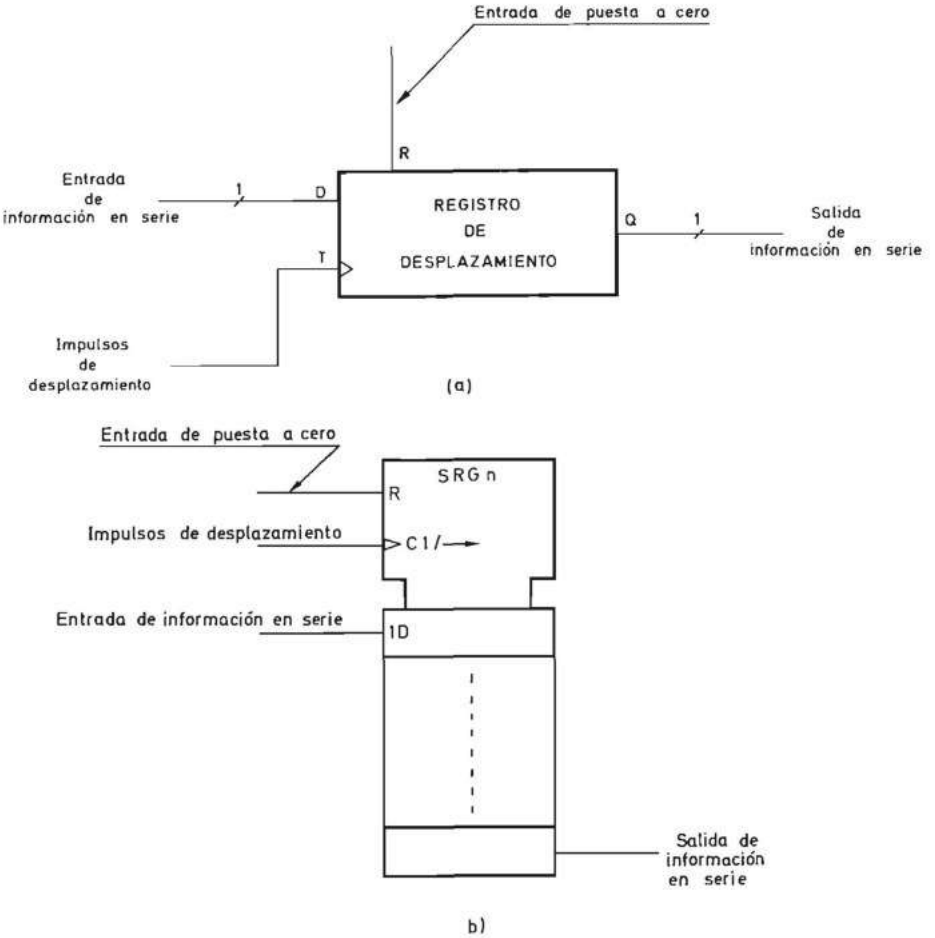


FIGURA 6.218.—**Símbolos lógicos correspondientes al registro de desplazamiento de la figura 6.217: *a*) No normalizado; *b*) Normalizado.**

R común a todos ellos. En la figura 6.218*a* y *b* se representan los símbolos lógicos correspondientes a este sistema. El símbolo normalizado utiliza el indicativo SRG [registro de desplazamiento (shift register)] y posee las señales siguientes:

R : Entrada de puesta a cero asíncrona común a todos los biestables.
$C1/\rightarrow$: Entrada de impulsos que introducen la información de la entrada serie en el primer biestable ($C1$) y desplazan la del resto hacia la derecha ($\rightarrow$).
$1D$: Entrada del primer biestable.

Al igual que se vio con los contadores, en los registros de desplazamiento existen diversos conceptos con variantes alternativas que han dado lugar a diferentes bloques funcionales integrados.

Por ejemplo, se puede lograr un registro de desplazamiento con entradas en paralelo asíncronas si se dota a los biestables de la figura 6.217 de entradas de puesta a uno asíncronas S y se realiza el montaje de la figura 6.219. Cada terminal S y R se conecta a la salida de una puerta Y de dos entradas. Una de las entradas de estas puertas, que se conectan todas entre sí y a un terminal único, se denomina de inhibición de entrada en paralelo ($\bar{I}_E$). A cada biestable se asigna un terminal de entrada en paralelo que se conecta directamente a la otra entrada de la puerta Y unida a la entrada de puesta a uno S y a través de un inversor a la entrada de la puerta Y unida a la entrada de puesta a cero R. Con los conocimientos adquiridos sobre biestables en el apartado 6.3.2 resulta fácil comprender que cuando $\bar{I}_E$ se pone en nivel uno se introduce en paralelo en los biestables la información presente en las entradas en paralelo asíncronas. En la figura 6.220*a* y *b* se representan los símbolos lógicos correspondientes al registro de la figura 6.219. El terminal de inhibición de entrada ($\bar{I}_E$) puede ser considerado como un terminal de control de entrada de información en paralelo que actúa por nivel. Por ello en la figu-

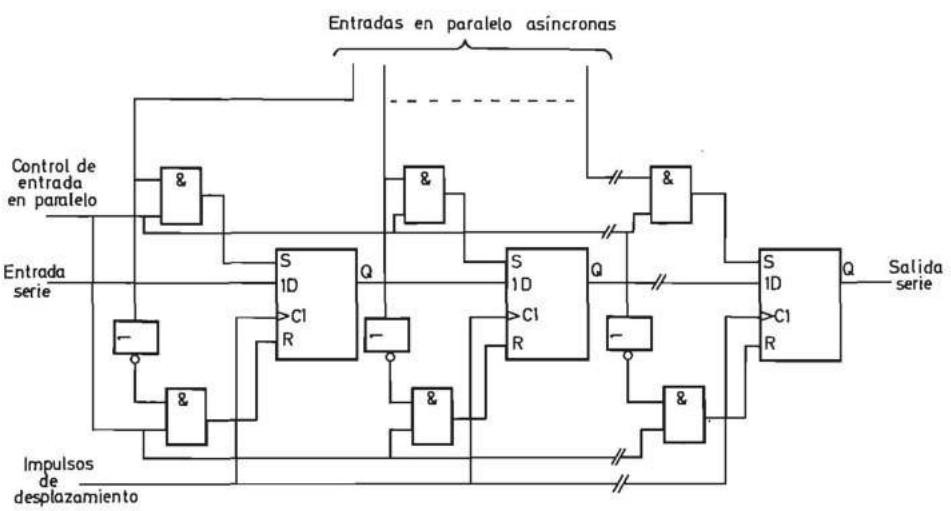


FIGURA 6.219.—**Esquema de un registro de desplazamiento con entradas en paralelo asíncronas.**

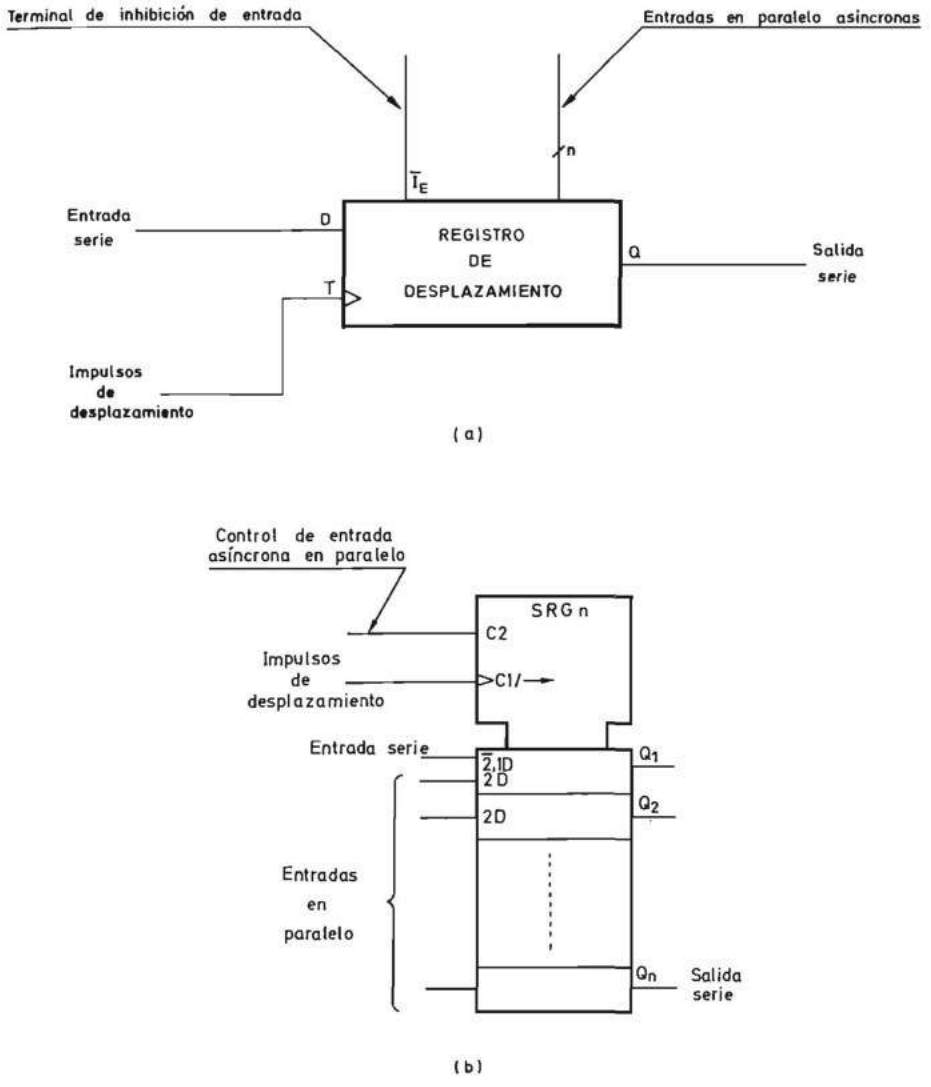


FIGURA 6.220.— **Símbolos lógicos correspondientes al registro de desplazamiento de la figura 6.219: *a*) No normalizado; *b*) Normalizado.**

ra 6.220*b* recibe la denominación *C*2, que se combina con la de las entradas de información en paralelo (2*D*). La entrada serie se identifica con los símbolos $\overline{2}$, 1*D* que indican que la información se desplaza al aplicar impulsos de desplazamiento en la entrada *C*1/→(1) si la entrada *C*2 se encuentra en nivel cero ($\overline{2}$).

Se realiza un registro de desplazamiento con entradas en paralelo síncronas si al circuito de la figura 6.217 se le añaden multiplexores de dos canales como los indicados en la figura 6.221. La entrada *D* de cada biestable se conecta a la sali-

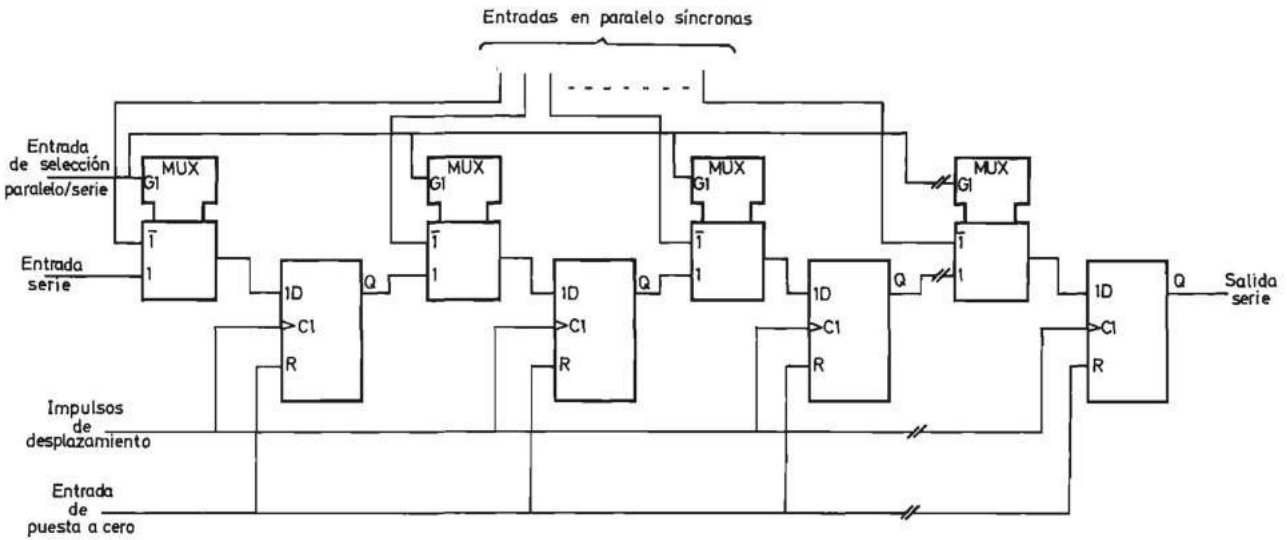


FIGURA 6.221.—**Esquema de un registro de desplazamiento con entradas en paralelo síncronas.**

da de un multiplexor de dos canales de entrada. Una de las entradas de éste se lleva al exterior como entrada en paralelo y la otra se une a la salida Q del biestable anterior. Las entradas de selección de los multiplexores se unen todas entre sí y el terminal único así obtenido constituye una variable de selección de modo de funcionamiento porque, según se encuentre en cero o en uno, los impulsos aplicados en la entrada $T(C)$ de los biestables hacen que la información se desplace en el registro o, por el contrario, que en él se introduzca la información presente en las entradas en paralelo. En la figura 6.222 se representan los símbolos lógicos correspondientes al registro de desplazamiento de la figura 6.221.

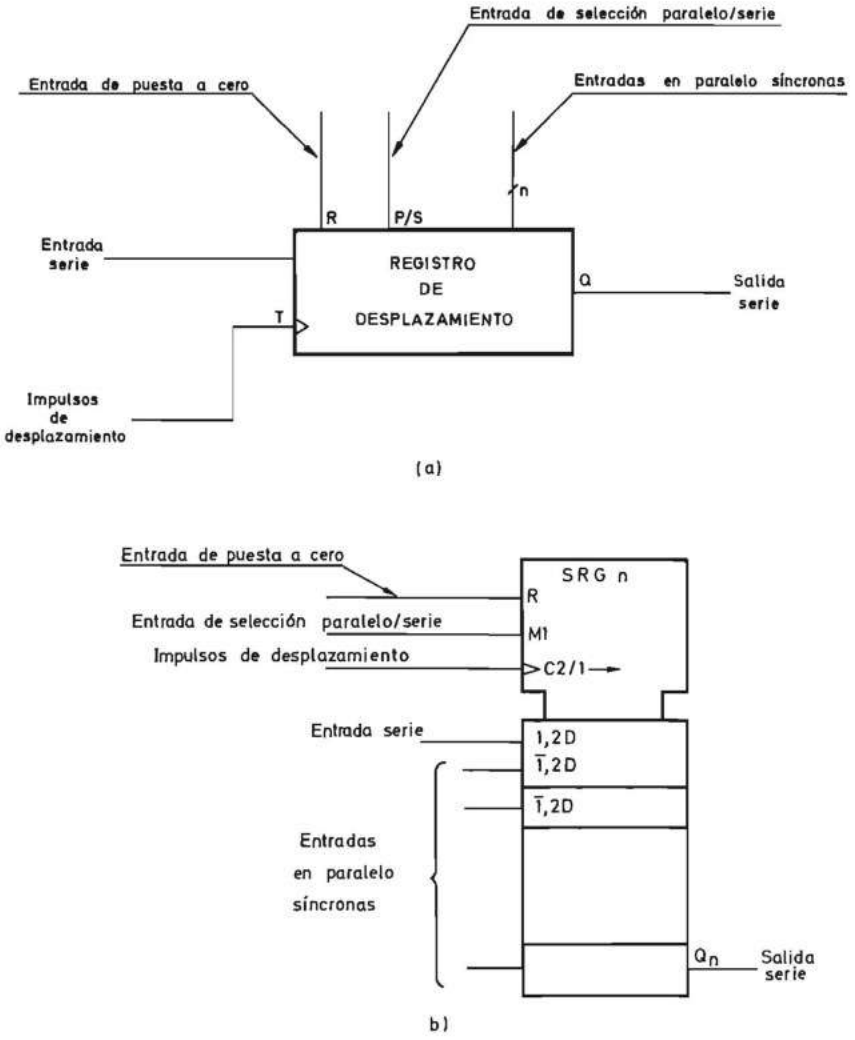


FIGURA 6.222.—**Símbolos lógicos correspondientes al registro de desplazamiento de la figura 6.221: *a*) No normalizado; *b*) Normalizado.**

En el símbolo normalizado se indican los siguientes terminales:

R : Entrada de puesta a cero común a todos los biestables.
$M1$: Entrada de selección paralelo/serie.
$C2/1\rightarrow$: Entrada de impulsos que producen el desplazamiento si $M1 = 1$ (1).
$1,2D$: Entrada serie cuya información se introduce en el primer biestable si $M1 = 1$ (1) y se aplica un flanco activo (en este caso el de subida) en $C2/1\rightarrow(2D)$.
$\overline{1},2D$: Entrada en paralelo cuya información se introduce en el biestable correspondiente si $M1 = 0$ ($\overline{1}$) y se aplica un flanco activo (en este caso el de subida) en $C2/1\rightarrow(2D)$.
Q_n : Salida serie.

Es posible dotar también a los registros de desplazamiento de una entrada de inhibición cuyo paso a un determinado estado activo inhiba la acción de los impulsos de tal manera que, al aplicar éstos, la información permanezca inalterada en el interior del registro. En la figura 6.223 se representa un registro de desplazamiento realizado con biestables JK que posee entrada de inhibición. En lugar de conectar directamente las salidas Q y $\overline{Q}$ de cada biestable a las entradas J y K respectivamente del siguiente, lo hacen a través de puertas Y de dos entradas. Las segundas entradas de estas puertas se unen todas entre sí y constituyen un terminal de inhibición. En efecto cuando este terminal se pone a cero, pasan también a cero las entradas J y K de todos los biestables y al aplicar impulsos de desplazamiento, el contenido de los biestables permanece inalterado. En la figura 6.224 se representan los símbolos lógicos correspondientes al registro de desplazamiento de la figura 6.223. El lector debe analizar los indicativos de cada uno de los terminales. Se recomienda igualmente al lector que, como ejercicio, dibuje el esquema de un registro de desplazamiento con biestables D que posea entrada de inhibición.

Es posible combinar la inhibición con la entrada en paralelo síncrona y tener un bloque funcional que posea ambas funciones. La ampliación de la capacidad de los bloques funcionales registros de desplazamiento representados en las figu-

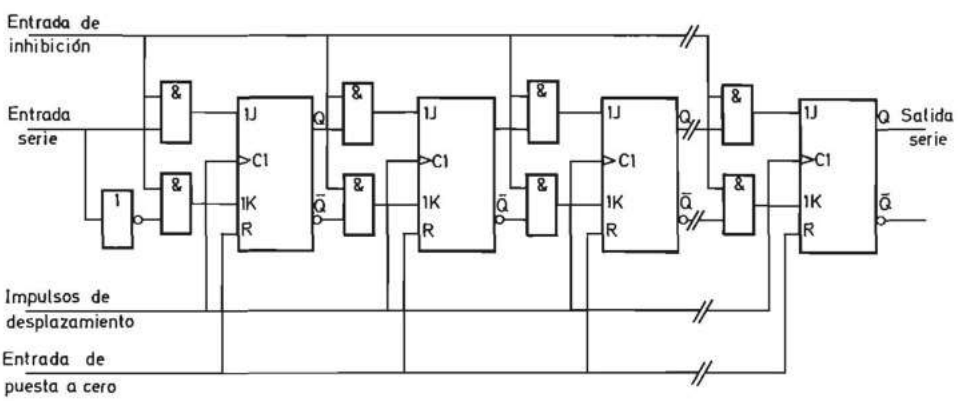


FIGURA 6.223.—**Esquema de un registro de desplazamiento con entrada de inhibición.**

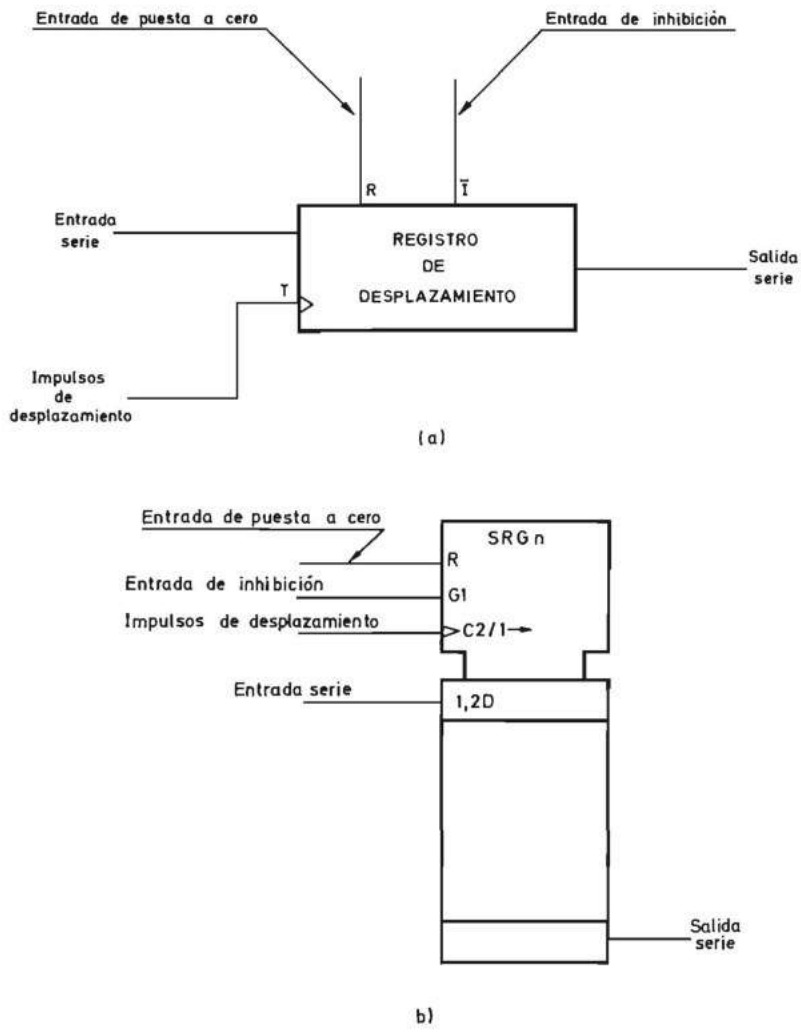


FIGURA 6.224.—**Símbolos lógicos correspondientes al registro de desplazamiento de la figura 6.223: *a*) No normalizado; *b*) Normalizado.**

ras 6.217 a 6.224 se realiza sin más que acoplar la entrada serie de uno a la salida de otro y unir entre sí el resto de los terminales. Como ejemplo en la figura 6.225 se representa un registro de desplazamiento resultante de acoplar dos bloques idénticos al de la figura 6.224.

Los registros representados en la figura 6.217 a 6.224 desplazan la información de izquierda a derecha. Es posible configurar un registro en el que la información se desplace en sentido contrario. En la figura 6.226 se representa un registro de desplazamiento hacia la izquierda realizado con biestables *D*. Para ello la entrada

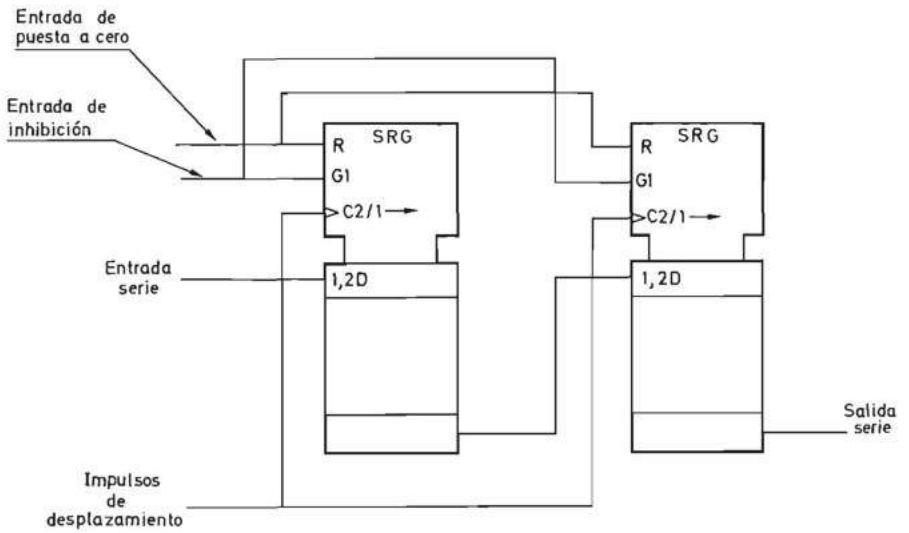


FIGURA 6.225.—**Registro de desplazamiento realizado mediante el acoplamiento de dos bloques funcionales.**

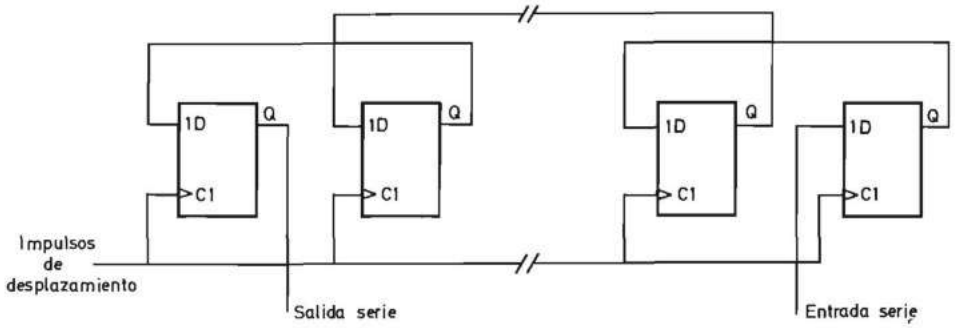


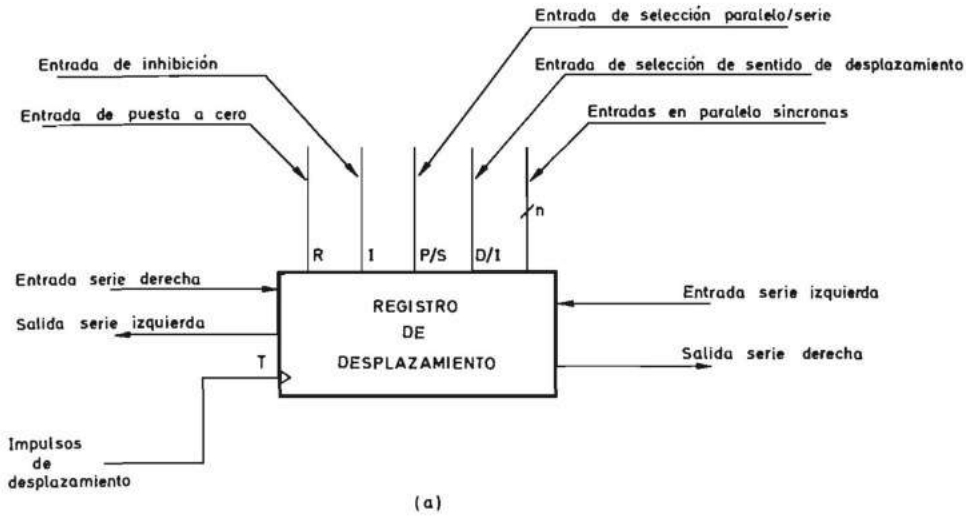
FIGURA 6.226.—**Esquema de un registro de desplazamiento hacia la izquierda.**

D de cada biestable se conecta a la salida *Q* del siguiente en lugar de a la del anterior.

Es posible realizar un registro de desplazamiento reversible en el que la información se desplace hacia la izquierda o hacia la derecha según el estado en que se encuentre una variable de modo de funcionamiento que puede denominarse *D/I*.

En la figura 6.227 se representa el símbolo lógico de un registro de desplazamiento reversible que posee:

— Una entrada *M*1 de selección de sentido de desplazamiento.



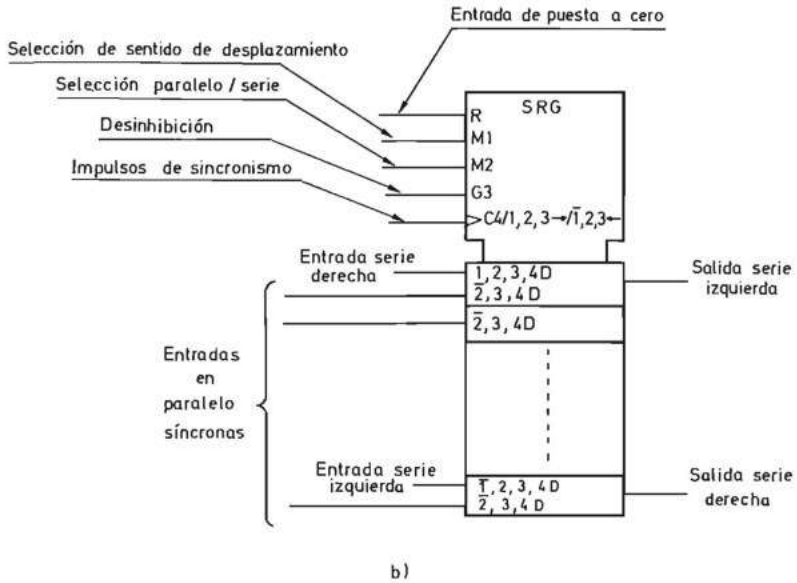


FIGURA 6.227.—**Símbolos lógicos correspondientes a un registro de desplazamiento reversible: *a*) No normalizado; *b*) Normalizado.**

— Una entrada *M*2 de selección de la entrada en paralelo o el desplazamiento.
— Una entrada *G*3 de desinhibición.
— Una entrada de impulsos de sincronismo que recibe la denominación *C*4/1, 2, 3 → / $\overline{1}$, 2, 3 → que indica, mediante el dígito 4, que los impulsos aplicados a ella actúan sobre el estado de los biestables. La forma de actuar depende de

$M1$, $M2$ y $G3$. Los indicativos 1, 2, 3 → especifican que los impulsos producen el desplazamiento de la información hacia la derecha si $M1 = 1$, $M2 = 1$ y $M3 = 1$; por el contrario los indicativos $\overline{1}$, 2, 3 → especifican que la información se desplaza hacia la izquierda si $M1 = 0$ y $M2 = M3 = 1$.

— Una entrada serie hacia la derecha que se denomina, 1, 2, 3, 4*D* porque su información se introduce en el primer biestable cuando $M1 = M2 = M3 = 1$ y se aplica un flanco activo en $C4$.

— Una entrada serie hacia la izquierda que se denomina, $\overline{1}$, 2, 3, 4*D* porque su información se introduce en el último biestable cuando $M1 = 0$, $M2 = M3 = 1$ y se aplica un flanco activo en $C4$.

— Terminales de entrada en paralelo que se denominan $\overline{2}$, 3, 4*D* porque su información se introduce en los biestables correspondientes si $M2 = 0$ y $M3 = 1$ y se aplica un flanco activo en $C4$.

— Un terminal de salida serie hacia la derecha y otro de salida serie hacia la izquierda.

En las diferentes tecnologías existen circuitos integrados de escala de integración media que poseen uno o más de los modos de operación descritos.

6.3.5.4.4.2 Aplicaciones de los registros de desplazamiento. Las aplicaciones de los registros de desplazamiento son numerosas. La mayoría de ellas hacen que estos registros formen parte de sistemas digitales más complejos de aplicación general que han sido realizados en circuitos de gran escala de integración (LSI).

A continuación se indican las aplicaciones más destacadas:

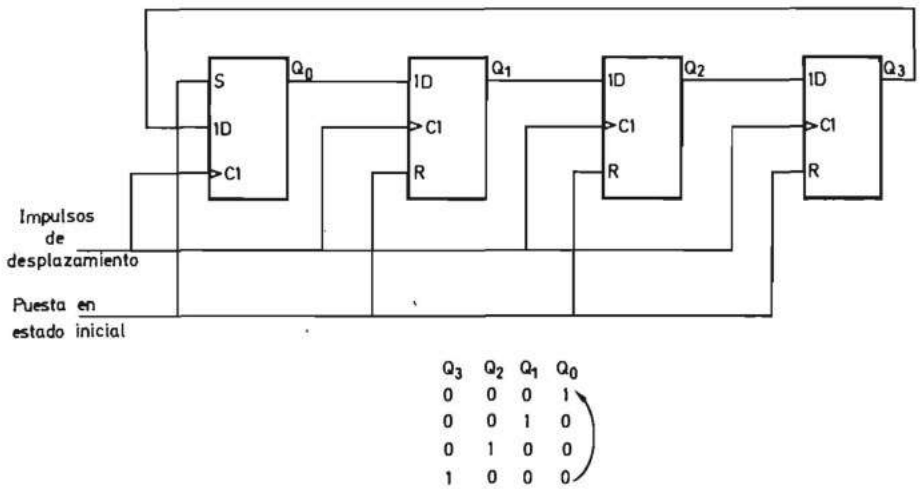
a) Contadores en códigos especiales.

Si inicialmente se pone en estado uno un solo biestable (por ejemplo, el primero de la izquierda), y se conecta la salida del último biestable a la entrada del primero, se obtiene un contador en anillo cuyo esquema y código de contaje se representan en la figura 6.228. La puesta en estado inicial se realiza mediante una variable *P* unida a la entrada de puesta a uno del primer biestable de la izquierda y la de puesta a cero de todos los demás. Un contador en anillo presenta la ventaja de que no necesita decodificador para detectar sus estados internos pero el inconveniente de poseer un biestable por cada estado interno.

Otro tipo de contador realizable con un registro de desplazamiento es el que efectúa el contaje en un código progresivo Johnson (ver apartado 1.6.1) que es continuo y cíclico. En la figura 6.229 se representa el esquema correspondiente realizado con biestables *J-K*. Las salidas del último biestable se conectan invertidas a las entradas del primero, e inicialmente se ponen todos los biestables en estado cero. El lector deducirá fácilmente que el código de contaje es el indicado en la propia figura 6.229. Los contadores en código Johnson tienen un número de estados internos igual al de dos veces el de biestables y presentan, con respecto a los contadores BCD natural y binario natural, una más fácil decodificación.

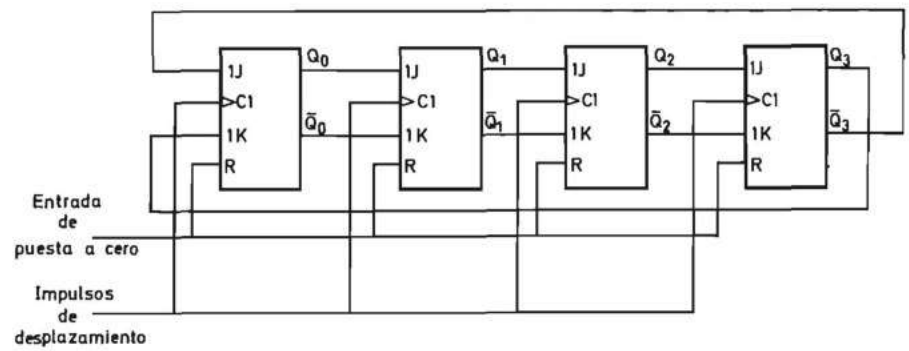
b) Conversión paralelo-serie.

La información se transmite en serie para reducir el número de canales necesa-



Q_3	Q_2	Q_1	Q_0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0

FIGURA 6.228.—**Esquema de un contador en anillo realizado con un registro de desplazamiento.**



Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1
1	1	1	0
1	1	0	0
1	0	0	0

FIGURA 6.229.—**Esquema de un contador en código Johnson realizado con un registro de desplazamiento.**

rios pero, por el contrario, se procesa en paralelo tal como se estudiará en el capítulo 8. Por ello, se hace necesaria la conversión de paralelo a serie, lo cual se realiza mediante un registro de desplazamiento en el que se introduce la información en paralelo, y se transmite en serie (fig. 6.230). Los registros de desplazamiento

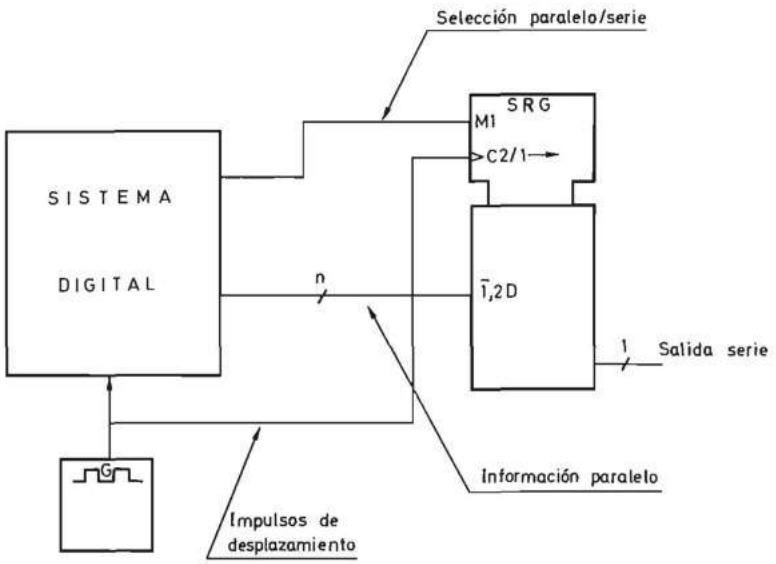


FIGURA 6.230.—**Aplicación de un registro de desplazamiento a la conversión paralelo-serie.**

forman parte de todos los circuitos de gran escala de integración desarrollados para la transmisión de información en serie [MAND 95] [STON 83].

c) Conversión serie-paralelo.

Este proceso, inverso al descrito en el apartado anterior, es el usual en la recepción de información transmitida en serie y que debe ser introducida en paralelo en un sistema digital. Se realiza mediante un registro de desplazamiento en el que se introduce la información en serie y se lee en paralelo (fig. 6.231). Por ello, los

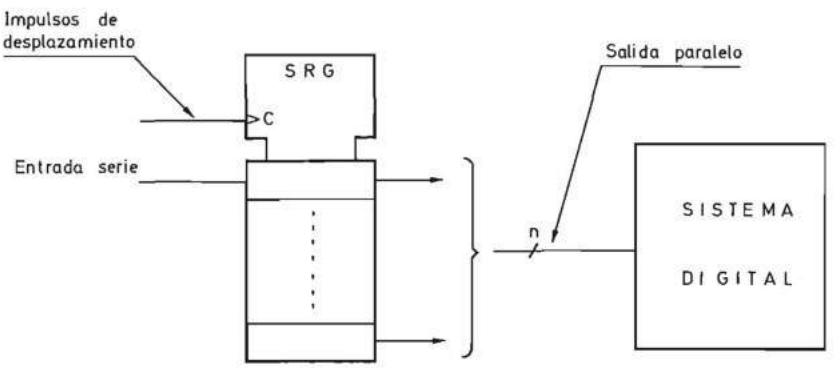


FIGURA 6.231.—**Aplicación de un registro de desplazamiento a la conversión serie-paralelo.**

registros de desplazamiento forman también parte de los circuitos de gran escala de integración desarrollados para la recepción de información en serie que se estudia en el tomo II [STON 83].

6.3.5.4.5 Registros de entrada y salida en paralelo. Los registros de entrada y salida en paralelo cuyo fundamento básico se estudió en el apartado 6.3.5.2.1 constituyen también bloques funcionales de aplicación general. Se utilizan como memoria temporal de datos de entrada y resultados parciales en los procesadores digitales y suelen estar constituidos por biestales *D*.

Como se vio anteriormente, los bloques funcionales constituidos por un registro realimentado como los contadores, registros de desplazamiento, etc., han de estar constituidos por biestables activados por flancos. Por el contrario, los registros de entrada y salida en paralelo pueden estar formados por biestables sincronizados por niveles o por flancos.

Los registros formados por biestables activados por niveles presentan la ventaja de la mayor sencillez del circuito. Su característica funcional más importante es la de que el impulso que se aplica a la entrada de control *C* de los biestables ha de situarse en el interior del intervalo durante el cual la información está estable en las entradas *D*. Esto se especifica mediante el cronograma de la figura 6.232 en el que la permanencia de las entradas *D* en un cierto estado durante un determinado intervalo de tiempo se indica mediante una línea continua.

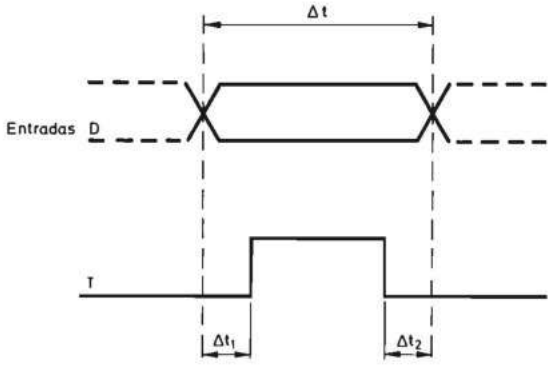


FIGURA 6.232.—**Diagrama de secuencia temporal de las señales aplicadas a un registro de entrada paralelo formado por biestables activados por niveles.**

En todo caso uno de los dos intervalos Δt_1, o Δt_2 puede ser nulo pero nunca negativo.

Los biestables sincronizados por flancos son más complejos pero, por el contrario, solamente es necesario aplicar en su entrada *C* un flanco activo en el interior del intervalo durante el cual la información está estable en las entradas *D*. En la figura 6.233*a* se representa un ejemplo en el que la información presente en las

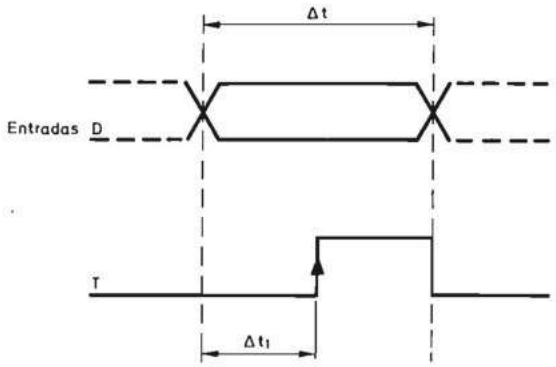


FIGURA 6.233.—**Diagrama de secuencia temporal de las señales aplicadas a un registro de entrada paralelo formado por biestables activados por flancos.**

entradas D durante el intervalo Δt se introduce en el registro mediante el flanco de subida aplicado a C en el medio del citado intervalo. Pero en el caso de que el flanco final de la señal aplicada a C (en este caso el de bajada) sea el que produzca el cambio de información en las entradas D, tal como se indica en la figura 6.233*b*, puede ser utilizado para transferir la información al registro.

En las figuras 6.234*a* y *b* se representa el símbolo lógico de un registro de entrada y salida en paralelo activado por flancos de subida. Además de las entradas en paralelo D, las salidas en paralelo Q, y la entrada de impulsos de transferencia T, el registro posee una entrada de inhibición $\overline{I}$ ($G1$) que se ha supuesto activa con un cero lógico. Para que los impulsos de transferencia hagan que la información se memorice en el registro es necesario que $\overline{I}$ ($G1$) se encuentre en nivel uno.

En la figura 6.234*c* se representa el cronograma correspondiente. Se supone que las señales de control del registro se obtienen a partir de un generador de impulsos que se aplica directamente a la entrada C. Además, durante un período del generador se hace aparecer una cierta información en las entradas D del registro ($1,2D$) y se coloca la entrada $\overline{I}$ ($G1$) en nivel uno. Si, tal como indica el cronograma de la figura 6.234*c*, los cambios de $G1$ y de la información en D se hacen a partir de los flancos de bajada del generador, pueden ser ellos mismos los que provoquen la entrada de información en el registro.

En la figura 6.235 se representa una aplicación típica de los registros de entrada y salida en paralelo utilizados como memoria temporal. En el instante en el que se quiere visualizar el contenido del contador, se aplica un impulso de transferencia a los registros de entrada y salida en paralelo que, en este caso, están realizados con biestables activados por niveles.

Los registros de entrada y salida en paralelo constituyen uno de los bloques funcionales utilizados en la realización de los procesadores digitales (ver tomo II).

6.3.5.4.6 Conjuntos de registros. Cuando se deben almacenar varias informaciones binarias simultáneamente se necesitan varios registros. Para ello se pueden utilizar registros de entrada y salida en paralelo independientes, pero esto pre-

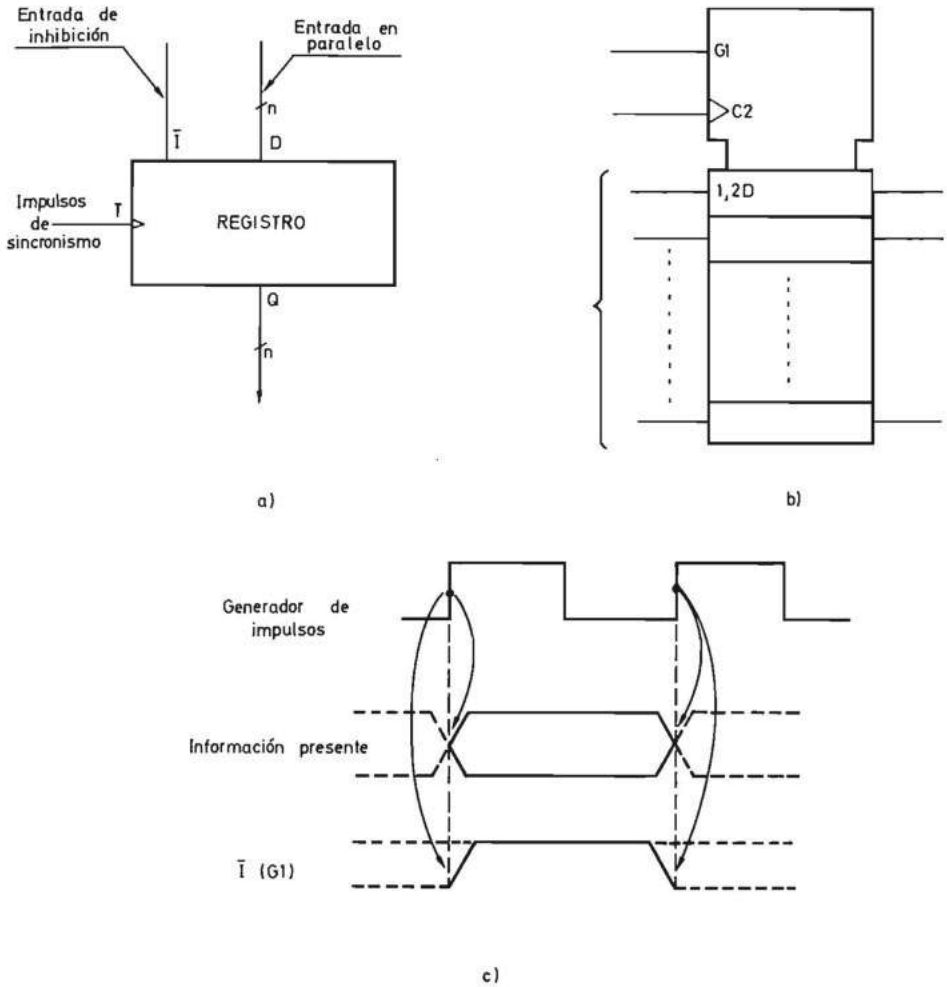


FIGURA 6.234.—**Registro de entrada y salida en paralelo formado por biestables activados por flancos de subida: *a*) Símbolo lógico no normalizado; *b*) Símbolo lógico normalizado; *c*) Diagrama de secuencia temporal de señales aplicadas a él.**

senta el inconveniente de complicar excesivamente el cableado. Por otra parte, si el almacenamiento se realiza en secuencia en lugar de simultáneamente, es posible agrupar los registros de tal manera que en cada instante sólo se seleccione uno para introducir información en él y otro para leer su contenido.

En la figura 6.236 se representa un conjunto de cuatro registros con entrada y salida en paralelo que tienen sus entradas de impulsos de transferencia y sus entradas *D* de información interconectadas entre sí. Cada registro posee una entrada de desinhibición *G*1. Cuando esta entrada se encuentra a nivel uno, el impulso de transferencia aplicado a *C*2 hace que la información presente en las entradas 1,2*D*

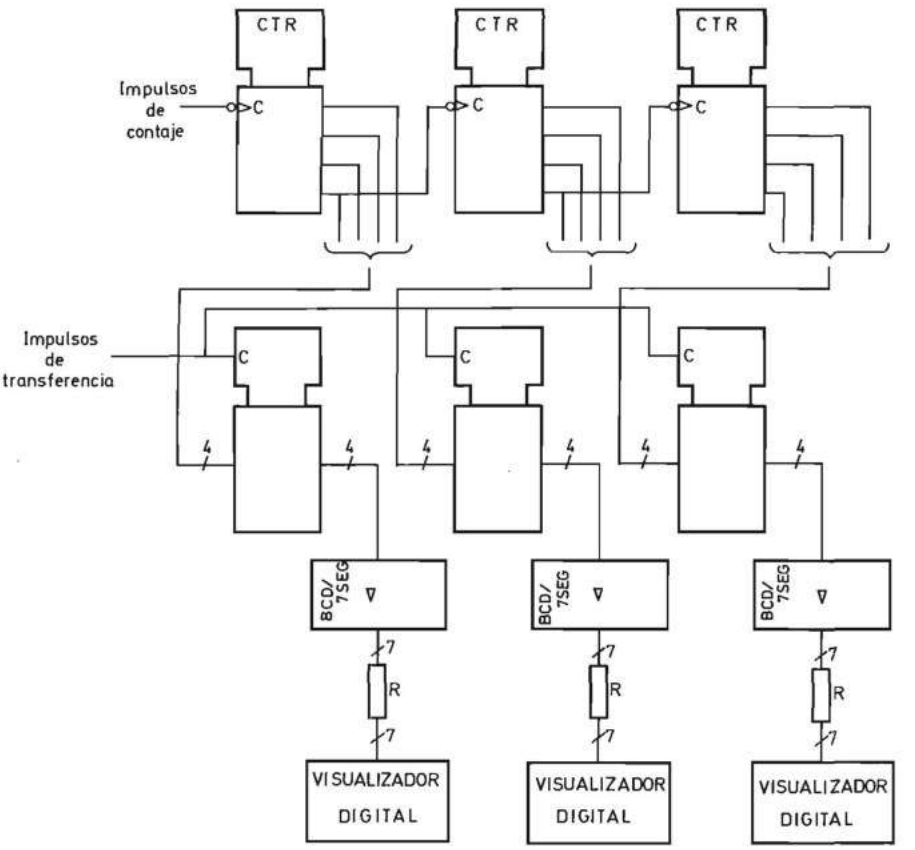


FIGURA 6.235.—**Esquema del circuito de aplicación de un registro de entrada y salida en paralelo a la visualización de un contador.**

se almacene en el registro correspondiente. Conectando todas las entradas *G*1 a la salida de un decodificador uno entre cuatro, se puede seleccionar, mediante las dos variables binarias de entrada de éste, cuál es el registro que memorizará la información presente en las entradas.

La salida de los registros se conecta a *n* multiplexores de cuatro canales que poseen dos variables de selección comunes. Mediante ellas se selecciona cuál es el registro cuyo contenido aparece a la salida.

Si se utilizan registros con salida de tres estados, se puede prescindir del multiplexor de la figura 6.236. En la figura 6.237 se representa el esquema correspondiente en el que las salidas de los cuatro registros se conectan entre sí, constituyendo una barra (bus) de información. Para seleccionar cuál es el registro que en cada instante presenta su información en la barra, se utiliza un decodificador de las dos variables de selección de salida, cuyas variables de salida se conectan a los terminales de desinhibición (*EN*) de cada registro. Ambos conjuntos de registros representados en las figuras 6.236 y 6.237 se pueden representar mediante los símbolos in-

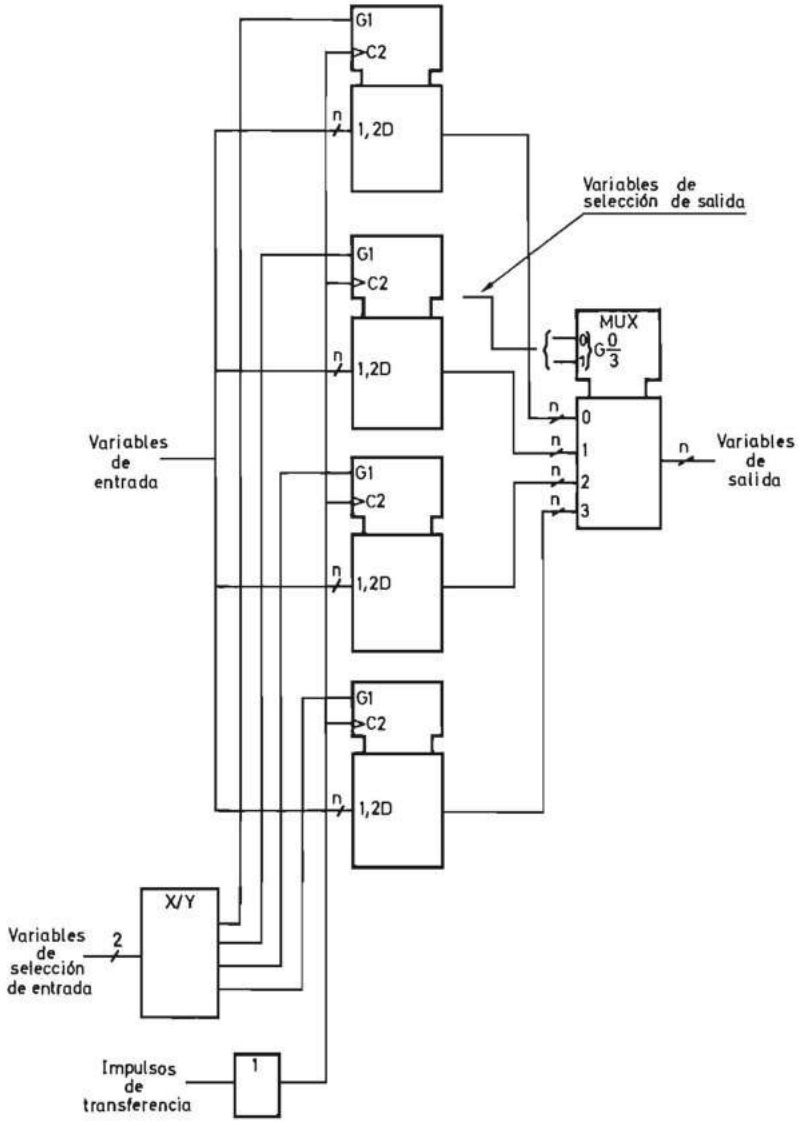


FIGURA 6.236.—**Esquema de un conjunto de registros con una única entrada y salida en paralelo realizado con un multiplexor de salida.**

dicados en la figura 6.238 en su versión no normalizada (*a*) y normalizada (*b*).

En la figura 6.238*b* las variables de selección de entrada reciben la denominación 1*A* y las de selección de salida 2*A*. Por ello, las entradas de información se denominan 1*A*, 4*D*, lo cual indica que se selecciona el registro en el que se introduce la información mediante los terminales $1A\ \frac{0}{3}$ y dicha acción se produce cuando

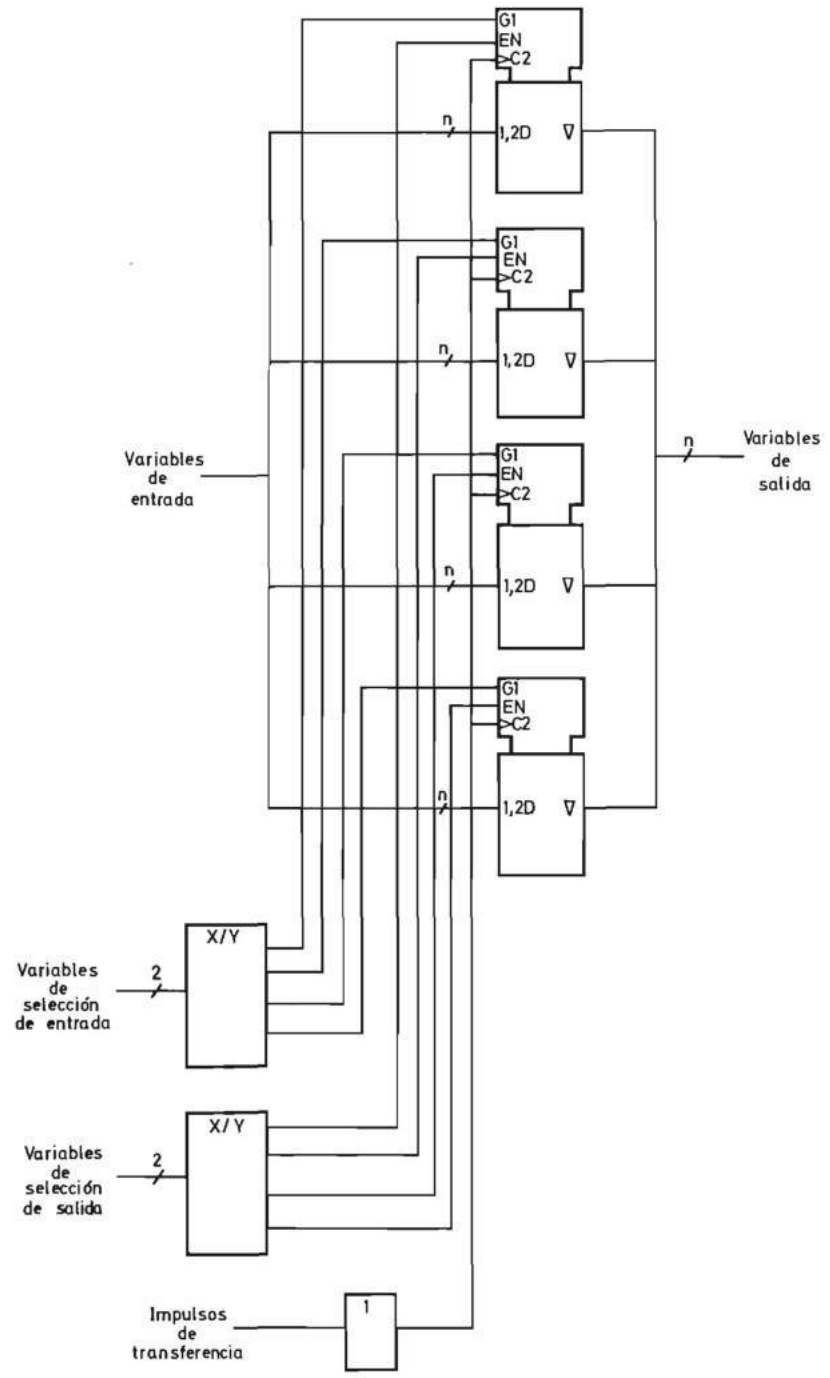


FIGURA 6.237.—**Esquema de un conjunto de registros con una única entrada y salida en paralelo realizado con una barra a la salida.**

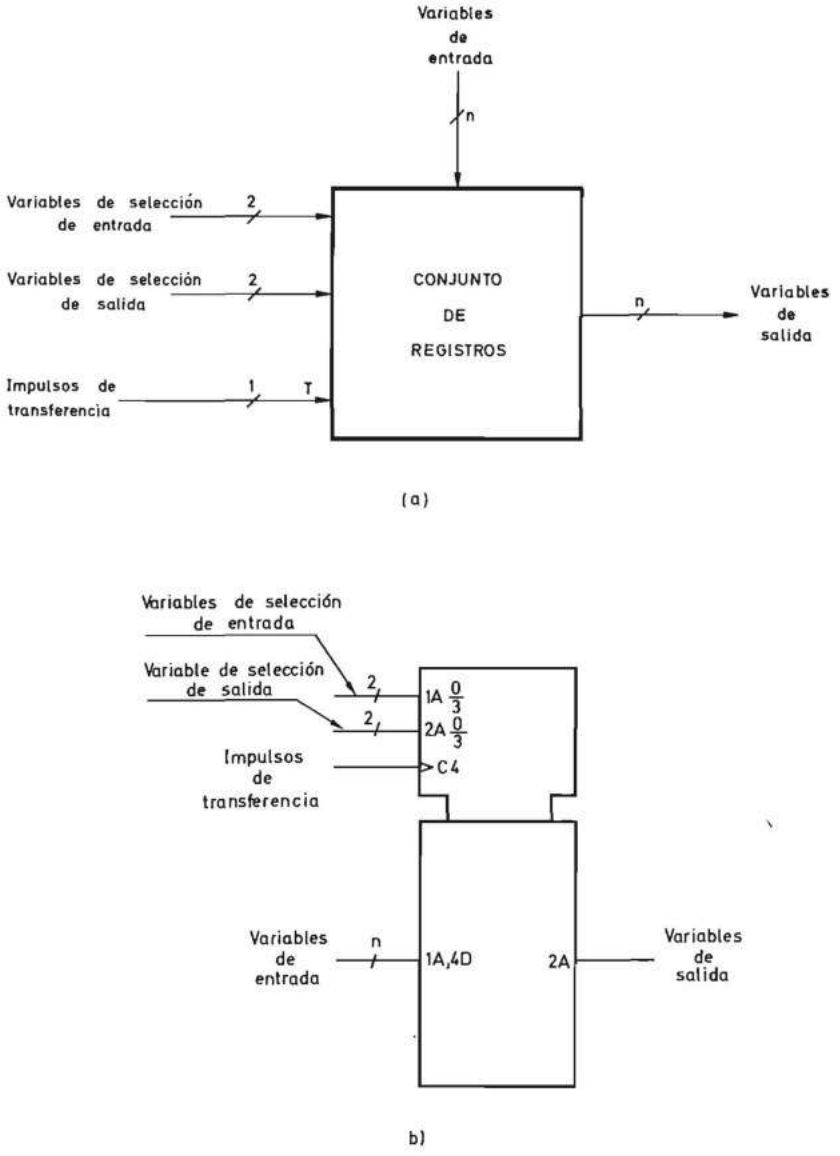


FIGURA 6.238.—**Símbolos lógicos correspondientes al conjunto de registros de las figuras 6.236 y 6.237: *a*) No normalizado; *b*) Normalizado.**

se aplica un impulso en *C*4 (4*D*). Las variables de salida llevan los símbolos 2*A* para indicar que mediante las variables de selección $2A\ \frac{0}{3}$ se elige el registro cuya información aparece en ellas.

En los registros de las figuras 6.236 y 6.237, la señal de control *C*2 actúa mediante el flanco de subida de los impulsos aplicados a ella, pero es posible también utilizar registros cuya señal de control actúe por niveles en lugar de por flancos.

Un conjunto de registros conectados de tal manera que no se tiene acceso simultáneamente a todos ellos, constituye una unidad de memoria. La importancia de las unidades de memoria hace que a ellas se dedique el capítulo 7 de este libro.

Por otra parte, los conjuntos de registros se utilizan principalmente en las unidades operativas de los procesadores y por ello sus aplicaciones se estudian en el tomo II de este libro.

6.3.5.5 Síntesis de los sistemas secuenciales síncronos de control. La síntesis de los sistemas secuenciales síncronos consiste en la obtención de un sistema físico cuyo estado interno evolucione de acuerdo con el diagrama de flujo.

Tal como se ha estudiado en apartados anteriores, un sistema secuencial síncrono está constituido por un registro, formado por biestables activados por flancos, realimentado a través de un circuito combinacional.

Cuando solamente se disponía en circuito integrado de puertas y biestables, el principal problema de la síntesis de los sistemas secuenciales síncronos consistía en la adecuada elección de la asignación de los estados de los biestables a cada uno de los estados internos del diagrama de flujo para minimizar el circuito combinacional. El desarrollo de la capacidad de integración ha relegado este problema a un segundo plano y ha propiciado nuevos métodos de síntesis que se estudian en sucesivos apartados.

6.3.5.5.1 Clasificación de los sistemas secuenciales síncronos. Los sistemas secuenciales síncronos se pueden clasificar de acuerdo con los dos conceptos compatibles entre sí (tabla 6.21) que se indican a continuación:

a) Según la forma de realizar el circuito combinacional.

De acuerdo con la forma de realizar el circuito combinacional, los sistemas secuenciales síncronos pueden ser cableados o programables.

Reciben la denominación de sistemas secuenciales síncronos cableados aquellos cuyo sistema combinacional se realiza mediante un conjunto de puertas ade-

CLASIFICACION DE LOS SISTEMAS SECUENCIALES SINCRONOS	Según el tipo de circuito combinacional	Cableados
		Programables
	Según la arquitectura	No modulares
		Modulares
		Semimodulares

TABLA 6.21

cuadamente interconectadas. Para cambiar el grafo o diagrama de flujo, es necesario cambiar las puertas o el cableado entre ellas.

Ejemplo de sistema secuencial síncrono cableado es el analizado en el apartado 6.3.5.1.

Los sistemas secuenciales síncronos programables, que suelen recibir la denominación de microprogramables, son aquellos en los que el sistema combinacional es programable y ha sido realizado mediante algunas de las formas estudiadas en apartado 3.8.

La característica más importante de los sistemas secuenciales síncronos microprogramables es que se puede cambiar su diagrama de un flujo sin necesidad de modificar el cableado de su sistema combinacional.

b) Según la estructura física

De acuerdo con la estructura física, los sistemas secuenciales síncronos pueden ser no modulares, semimodulares o modulares. Un sistema secuencial síncrono no modular es aquel en el cual una elevación del número de variables de entrada o de salida implica una variación, no solamente de los elementos que lo componen, sino también de la interconexión entre los mismos. Los sistemas secuenciales síncronos no modulares pueden ser cableados o programables según el tipo de circuito combinacional utilizado en su realización. Un ejemplo de sistema secuencial síncrono no modular cableado es el de la figura 6.158 analizada en el apartado 6.3.5.

Un sistema secuencial síncrono modular es aquel en el cual se puede ampliar el número de variables de entrada o de salida sin más que añadir los elementos adecuados y sin necesidad de cambiar la conexión de los elementos que lo constituyen. Aunque los sistemas secuenciales síncronos modulares pueden ser también cableados o programables, los primeros carecen de interés práctico porque una elevación del número de variables de entrada o salida supone un cambio en la tabla de verdad del circuito combinacional difícil de realizar si éste es cableado. Al análisis de los sistemas secuenciales síncronos modulares se dedica el apartado 6.3.5.5.4.

Un sistema secuencial síncrono es semimodular cuando el número total de variables de entrada más el de salida es constante pero puede variar el de cualquiera de ellas de tal manera que la otra lo haga en sentido contrario. Constituyen una solución intermedia entre los no modulares y los modulares y a su análisis se dedica el apartado 6.3.5.5.5. Por las mismas razones indicadas antes para los modulares, los sistemas secuenciales síncronos semimodulares se combinan en la práctica con los circuitos combinacionales programables.

La compatibilidad de las dos clasificaciones de los sistemas secuenciales síncronos indicadas en la tabla 6.21 hace que su estudio no sea tarea fácil de organizar. En este libro se ha decidido estudiar en primer lugar los sistemas secuenciales síncronos cableados y programables sin tener en cuenta la problemática de la modularidad para seguidamente estudiar, a partir de ellos, los sistemas secuenciales síncronos modulares y semimodulares.

6.3.5.5.2 Síntesis de los sistemas secuenciales síncronos cableados

6.3.5.5.2.1 Síntesis de los sistemas secuenciales síncronos cableados con registros de entrada y salida en paralelo. Tal como se indica en el apartado 6.3.5.5.1, la disponibilidad de circuitos de escala de integración media ha sustituido en parte los criterios de minimización por los de sistematización en el diseño y utilización de bloques funcionales.

A continuación se exponen sendos métodos de codificación de los estados internos basados en estos criterios. Esta forma de realizar los sistemas secuenciales síncronos se utilizó con profusión cuando se comenzó a disponer de los primeros bloques funcionales integrados en escala de integración media. Aunque no se utilizan actualmente, su estudio resulta interesante por contribuir a que se comprenda mejor la problemática de la realización de los sistemas secuenciales síncronos.

a) Codificación en binario natural y utilización de un decodificador.

La disponibilidad de codificadores en circuito integrado hace interesante la codificación en binario natural y la utilización de los estados decodificados para la realimentación. La estructura de un autómata codificado de esta manera corresponde a la figura 6.239. La gran ventaja de este método es la de ser sistemático, lo que se hace patente cuando el número de estados internos y variables de entrada es elevado.

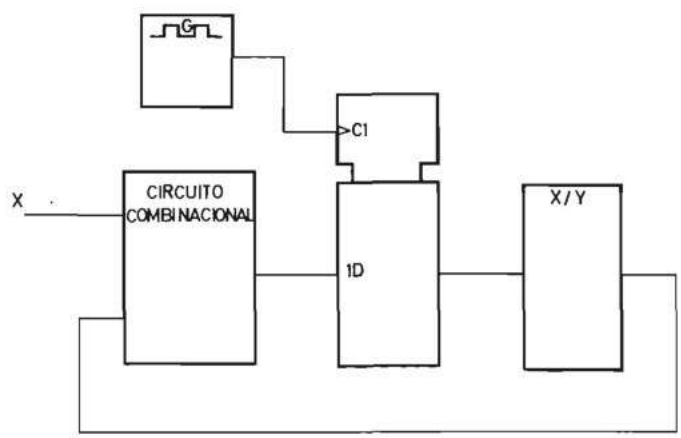


FIGURA 6.239.—**Diagrama de bloques de un sistema secuencial síncrono realizado con un registro de entrada y salida en paralelo y un decodificador.**

Para aclarar lo que se acaba de exponer, se aplica a continuación este método al sistema del ejemplo 6.4.

Ejemplo 6.5

Supóngase que en el diagrama de flujo de la figura 6.156 no están codificados los estados internos. Se tendrá de esta forma el diagrama de flujo de la figura 6.240.

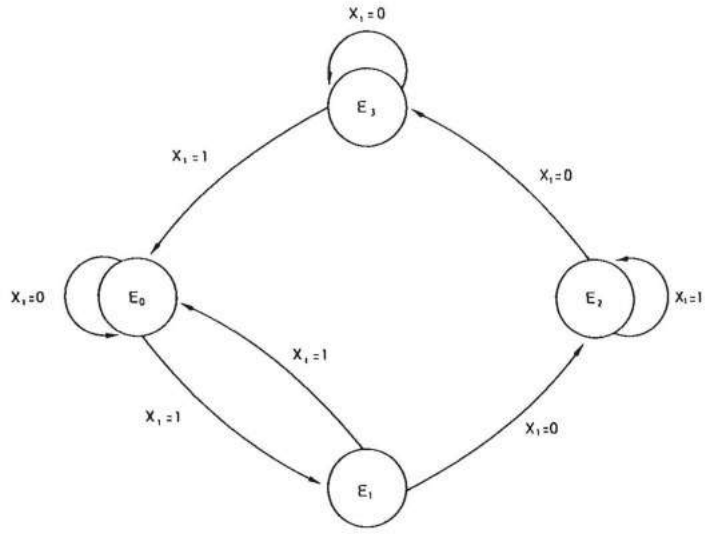


FIGURA 6.240.—**Diagrama de flujo correspondiente al sistema secuencial síncrono analizado en el ejemplo 6.4.**

Para la codificación de los cuatro estados internos son suficientes dos variables de estado interno Q_0 y Q_1. La asignación de los cuatro estados posibles de estas dos variables a los cuatro estados internos E_0 a E_3 es arbitraria. La complejidad del sistema combinacional obtenido no varía mucho de unas asignaciones a otras. En este ejemplo se realiza la asignación indicada en la tabla 6.22.

Para la realización de las variables Q_0 y Q_1 se utilizan biestables D activados por flancos, lo cual simplifica, como se verá, el método.

En primer lugar se comprueba en qué estados internos toma cada variable el estado uno. Por ejemplo, Q_0 toma el valor uno en los estados E_1 y E_3. Seguidamente se observa en qué condiciones de estado interno y estado de entrada el sistema ha de pasar a los estados E_1 o E_3 al aplicar un nuevo impulso de sincronismo.

Observando el diagrama de la figura 6.240 se comprueba que el sistema ha de pasar al estado E_1 si se encuentra en E_0 y x_1 toma el valor uno, lo cual viene indi-

	Q_1	Q_0
E_0	0	0
E_1	0	1
E_2	1	0
E_3	1	1

TABLA 6.22

cado por una flecha dirigida desde E_0 a E_1. De igual forma se deduce que el sistema ha de pasar a E_3 si se encuentra en E_2 y x_1 toma el valor cero, y ha de permanecer en E_3 si x_1 se encuentra en el estado cero. Por lo tanto, se deduce la ecuación de Q_{0t+1}:

$$Q_{0t+1} = E_0 x_1 + E_2 \bar{x}_1 + E_3 \bar{x}_1$$

De igual forma el lector puede deducir la ecuación de Q_{1t+1} y comprobar que el resultado es:

$$Q_{1t+1} = E_1 \bar{x}_1 + E_2 x_1 + E_2 \bar{x}_1 + E_3 \bar{x}_1 = E_1 \bar{x}_1 + E_2 (x_1 + \bar{x}_1) + E_3 \bar{x}_1 = \\ = E_1 \bar{x}_1 + E_2 + E_3 \bar{x}_1$$

Si se utilizan biestables D activados por flancos cuya ecuación característica es $Q_{t+1} = D_t$, se obtienen las ecuaciones de las entradas D de los biestables:

$$D_0 = E_0 x_1 + E_2 \bar{x}_1 + E_3 \bar{x}_1$$
$$D_1 = E_1 \bar{x}_1 + E_2 + E_3 \bar{x}_1$$

Realizando ambas ecuaciones con puertas NO-Y resulta:

$$D_0 = \overline{\overline{E_0 x_1} \; \overline{E_2 \bar{x}_1} \; \overline{E_3 \bar{x}_1}}$$

$$D_1 = \overline{\overline{E_1 \bar{x}_1} \; \overline{E_2} \; \overline{E_3 \bar{x}_1}}$$

En la figura 6.241 se representa el esquema completo del autómata obtenido.

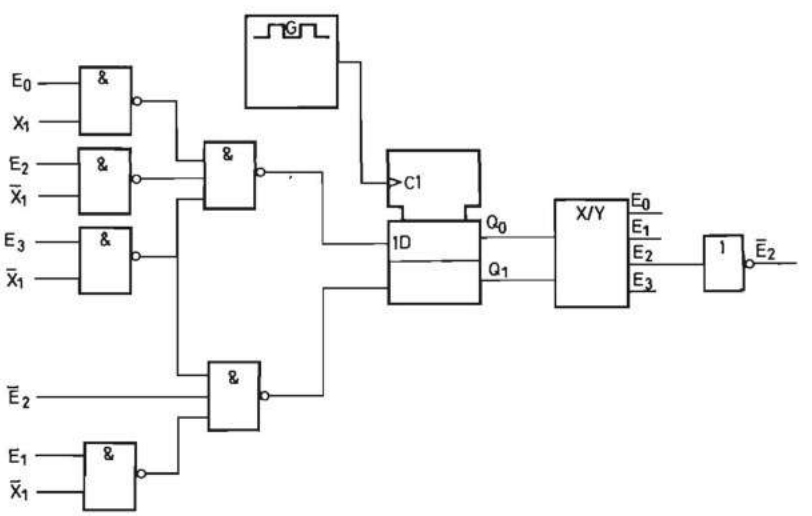


FIGURA 6.241.—**Esquema del sistema secuencial síncrono que ejecuta el diagrama de flujo de la figura 6.240 realizado con un registro y un decodificador.**

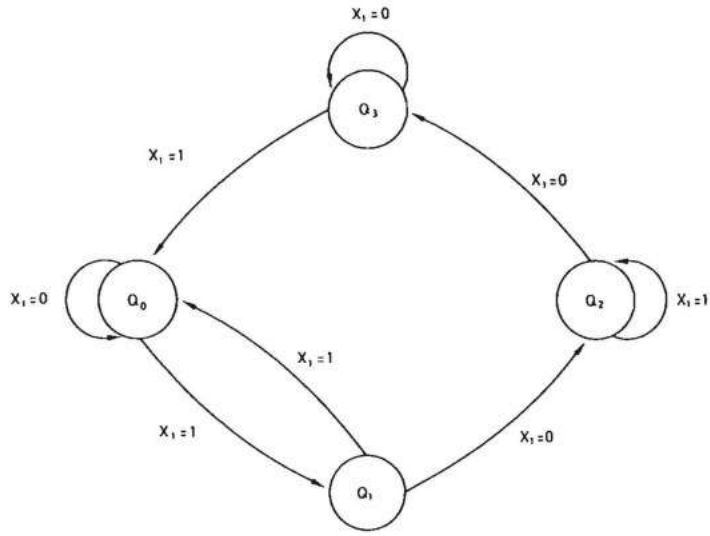


FIGURA 6.242.—**Diagrama de flujo del sistema secuencial síncrono analizado en el ejemplo 6.4 codificado mediante un código de uno entre *n*.**

b) Codificación mediante un código de uno entre *n*.

Otro procedimiento que simplifica la codificación y representa una solución todavía más sistemática que la del apartado anterior es la utilización de un biestable por estado interno. Si el número de estados internos es *n*, el número de biestables es también *n*. En cada estado interno del autómata uno solo de los *n* biestables se encuentra en estado uno.

Este método se divulgó cuando se dispuso, en un único circuito integrado, de registros de entrada y salida en paralelo que poseían cuatro o seis biestables *D* activados por flancos.

Ejemplo 6.6:

A continuación se aplica este método al sistema cuyo diagrama de flujo se representó en la figura 6.156. Para ello se asignan cuatro biestables Q_0 a Q_3 a los estados internos E_0 a E_3 respectivamente. Se obtiene de esta forma el diagrama de la figura 6.242 que es idéntico al de la figura 6.156 con la única diferencia de

	Q_0	Q_1	Q_2	Q_3
E_0	1	0	0	0
E_1	0	1	0	0
E_2	0	0	1	0
E_3	0	0	0	1

TABLA 6.23

que en el interior de cada círculo, correspondiente a un estado estable, se indica el biestable que toma el valor uno para dicho estado. En la tabla 6.23 se representa la asignación de estados de los diferentes biestables a cada estado interno.

De una forma similar a la descrita en el apartado anterior se observa en qué condiciones debe tomar el valor uno lógico cada biestable Q_0 a Q_3. Por ejemplo, el biestable Q_0 debe tomar el valor uno si Q_3 es uno y la variable de entrada x_1 también se encuentra en estado uno (fig. 6.242); de igual forma Q_0 ha de adoptar el estado uno si Q_1 es uno y la variable de entrada x_1 es uno. Además Q_0 ha de permanecer en uno si $x_1 = 0$. Por lo tanto, se deduce:

$$Q_{0t+1} = x_1 Q_1 + \bar{x}_1 Q_0 + Q_3 x_1$$

Como práctica, el lector puede escribir las ecuaciones de los demás biestables que resultan:

$$Q_{1t+1} = Q_0 x_1$$
$$Q_{2t+1} = Q_1 \bar{x}_1 + Q_2 x_1$$
$$Q_{3t+1} = Q_2 \bar{x}_1 + Q_3 \bar{x}_1$$

Estas ecuaciones se pueden realizar con puertas NO-Y, resultando:

$$Q_{0t+1} = \overline{\overline{x_1 Q_1} \; \overline{\bar{x}_1 Q_0} \; \overline{x_1 Q_3}}$$

$$Q_{1t+1} = \overline{\overline{x_1 Q_0}}$$

$$Q_{2t+1} = \overline{\overline{\bar{x}_1 Q_1} \; \overline{x_1 Q_2}}$$

$$Q_{3t+1} = \overline{\overline{\bar{x}_1 Q_2} \; \overline{\bar{x}_1 Q_3}}$$

En la figura 6.243 se representa el esquema completo del autómata realizado por este método.

En el instante inicial en que se da tensión al sistema todos los biestables han de quedar en estado cero, excepto uno. Definiendo uno de los estados internos como inicial, por medio de las entradas de puesta a cero y uno asíncronas se hace que el sistema quede inicialmente en ese estado.

En el sistema de la figura 6.243 se ha definido como estado inicial aquel en que Q_0 se encuentra en estado uno. Para poner inicialmente al sistema en ese estado se usan la entrada de puesta a uno (S) de Q_0 y las entradas de puesta a cero (R) de Q_1, Q_2 y Q_3.

6.3.5.5.2.2 Síntesis de los sistemas secuenciales síncronos cableados con contadores síncronos. En el apartado 6.3.5.4.1 se estudiaron los sistemas secuenciales síncronos y se vio que el diagrama de flujo más complejo posible permite las transiciones entre estados cualesquiera. Por otra parte, en el apartado 6.3.5.2.2 se analizó el diagrama de flujo de un contador (fig. 6.163) y en él se vio que es un sistema secuencial síncrono particular que no toma decisiones. Los diagramas de flujo

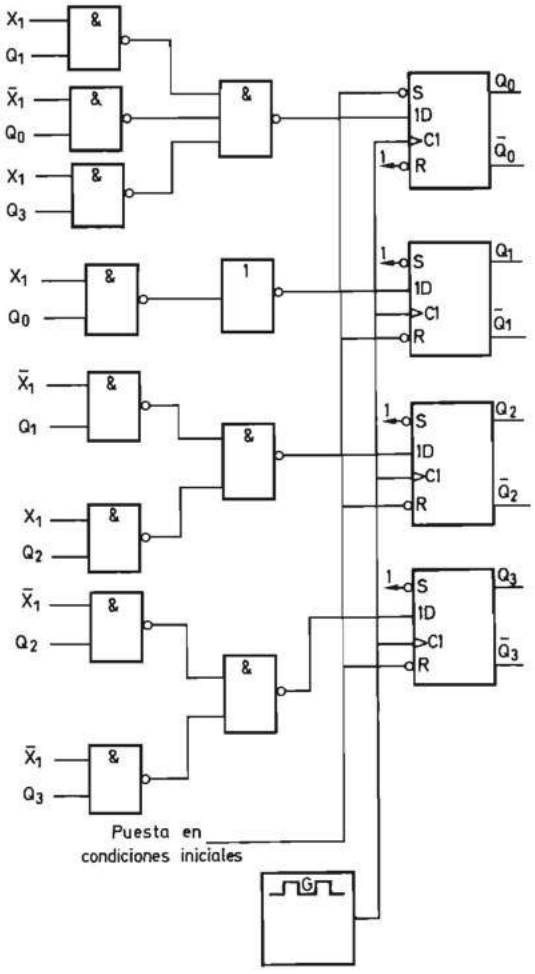


FIGURA 6.243.—**Esquema del sistema secuencial síncrono que ejecuta el diagrama de flujo de la figura 6.242.**

de los sistemas secuenciales síncronos más utilizados en la práctica son una combinación de ambos tal como se indica en la figura 6.244.

La evolución de estos sistemas entre estados internos se realiza de dos maneras:

a) Sin intervención de variables externas, de una forma similar a la de un contador.

b) Dependiendo del estado de una o más variables externas.

En la figura 6.245 se representa el diagrama de flujo de un sistema secuencial síncrono que responde a lo expuesto. El sistema secuencial síncrono que actúe de acuerdo con este diagrama pasa, por ejemplo, del estado E_4 al E_5 de forma inequívoca y, si se encuentran en E_2, puede pasar a E_3 o E_4 según la variable B esté res-

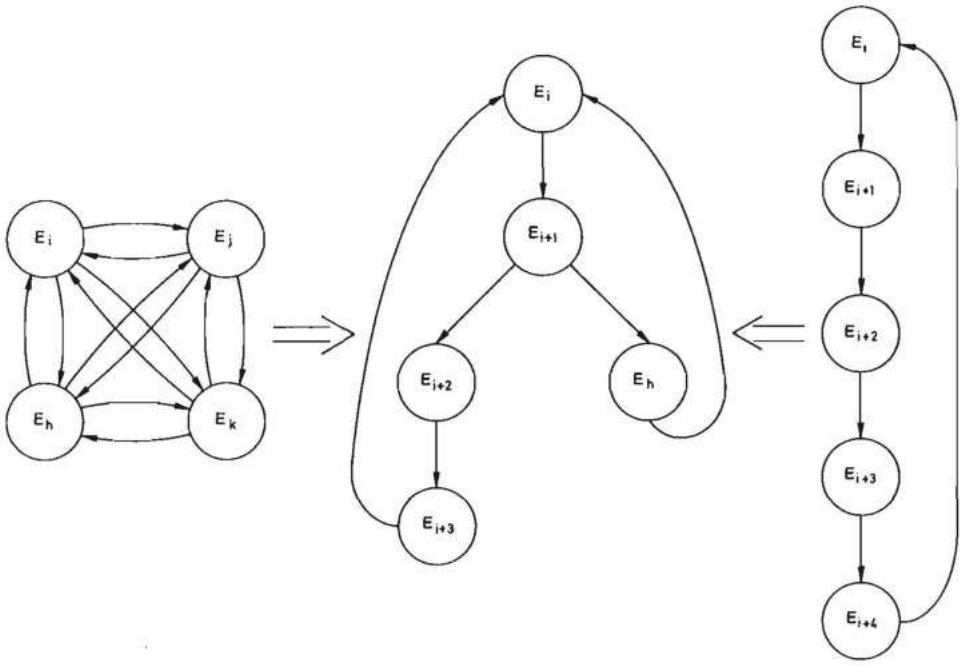


FIGURA 6.244.—**Diagramas de flujo de sistemas secuenciales síncronos.**

pectivamente en cero o en uno. Veremos seguidamente cómo se puede diseñar un sistema que actúe según el diagrama de la figura 6.245 mediante un contador síncrono con entradas en paralelo síncronas.

En la figura 6.246 se representa el símbolo lógico de un contador con entradas en paralelo síncronas, en sus versiones no normalizada (*a*) y normalizada (*b*). Este contador posee:

— Una entrada de puesta a cero (*R*) asíncrona activa con un uno lógico.

— Una entrada de impulsos de sincronismo ($C2/\overline{1}+$) activa con los flancos de subida.

— *n* entradas de información en paralelo (1,2*D*), una por cada biestable.

— Una entrada de modo de funcionamiento (*M*1). Si se encuentra en nivel cero, hace que se cuenten los impulsos de sincronismo en binario natural, y si se encuentra en nivel uno hace que dichos impulsos produzcan la entrada de la información presente en los terminales 1,2*D*.

El diagrama de bloques del sistema secuencial síncrono realizado mediante un contador se representa en la figura 6.247. El sistema combinacional recibe como entradas las salidas *Q* del contador y las variables externas y genera la señal $\overline{C}/P$ (*M*1) y las entradas *D* (1,2*D*) del contador.

Mediante el diagrama de flujo de la figura 6.245 se pueden exponer los conceptos básicos de la síntesis del sistema combinacional.

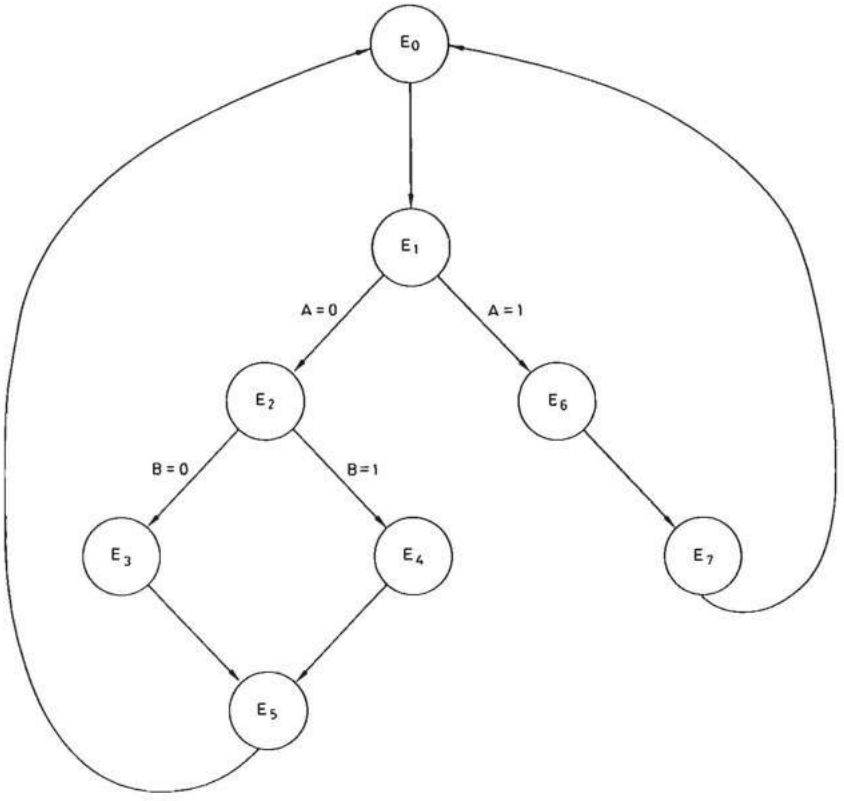


FIGURA 6.245.—**Ejemplo de diagrama de flujo típico de un sistema secuencial síncrono.**

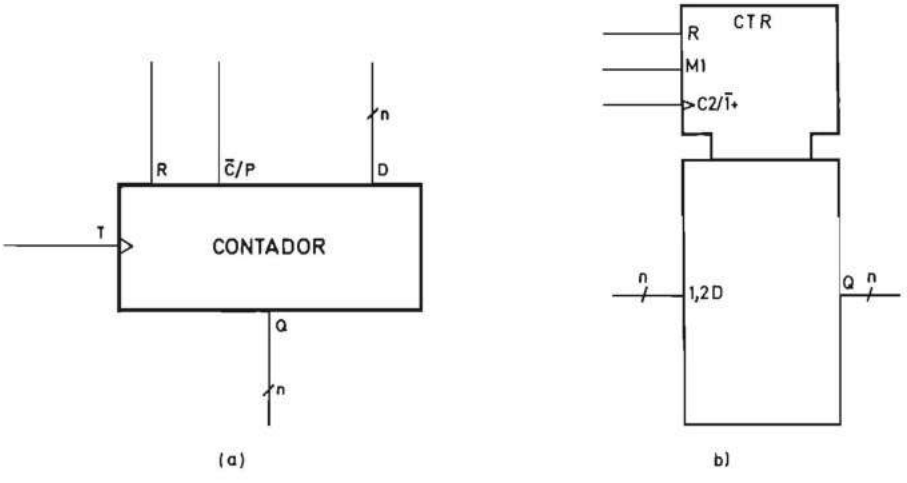


FIGURA 6.246.—**Símbolo lógico de un contador síncrono con entradas en paralelo síncronas: *a*) No normalizado; *b*) Normalizado.**

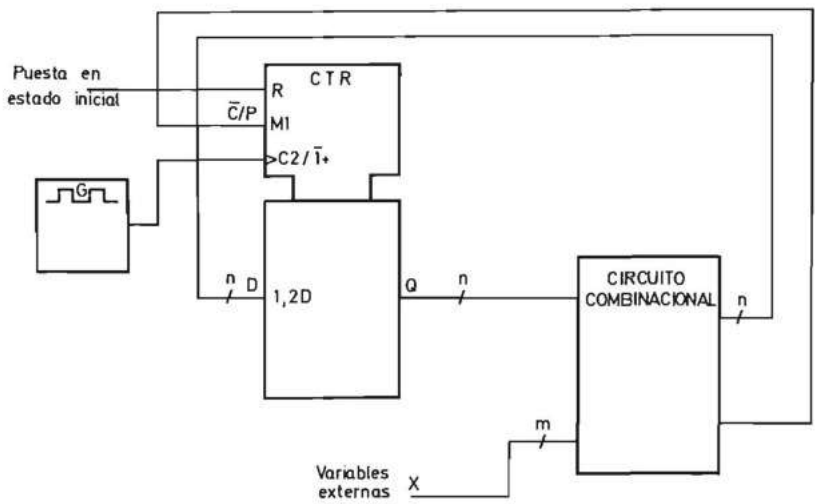


FIGURA 6.247.—**Diagrama de bloques de un sistema secuencial síncrono realizado con un contador síncrono con entradas en paralelo síncronas.**

En primer lugar se observa el número de estados del diagrama de flujo para deducir el número de biestables m que ha de tener el contador. El número m ha de ser tal que $2^m \geq n$, siendo n el número de estados del diagrama de flujo. El código de contaje puede ser cualquiera de los utilizados en los bloques funcionales disponibles en las diversas tecnologías, de los cuales los más usuales son el binario natural y el BCD natural. Si el número de estados es inferior a 9 se puede utilizar un contador en cualquiera de ambos códigos con un solo circuito integrado. Si el número de estados está comprendido entre 11 y 16 ambos inclusive, se puede realizar con un contador en binario natural (un solo circuito integrado) o en BCD natural (dos circuitos integrados).

La asignación de códigos binarios a los estados del grafo se debe realizar de tal forma que el número de saltos entre estados que no tengan asignados códigos consecutivos en binario natural o BCD natural sea mínimo. No existen reglas totalmente sistemáticas para realizar dicha asignación, pero esto no tiene mucha importancia porque, como se verá en el apartado 6.3.5.5.4, cuando el número de estados es elevado, se puede realizar el sistema combinacional mediante un bloque funcional integrado como los estudiados en el apartado 3.8.

Siempre que sea posible, a los estados consecutivos se les asignan números decimales consecutivos. Por ejemplo, E_0 y E_1 (fig. 6.245). Cuando se toma una decisión en función de una variable de estado interno, se debe asignar a uno de los estados el número contiguo al anterior. Siguiendo estas reglas se obtiene la asignación indicada en la figura 6.245.

Partiendo del grafo de la figura 6.245 se obtiene la tabla de verdad de la tabla 6.24. En ella se representan como variables independientes el estado del

contador en el instante t, antes de recibir el flanco activo, Q_{2t}, Q_{1t}, Q_{0t} y las variables externas A y B, y como variables dependientes el estado del contador en el instante $t + 1$, después de recibir el flanco activo y la variable $\overline{C}/P$. A continuación se deducen como ejemplo algunas líneas de la tabla. Si el sistema se encuentra en el estado $E_2(Q_{2t}\ Q_{1t}\ Q_{0t} = 010)$, el nuevo estado debe ser E_3 $(Q_{2t}\ Q_{1t}\ Q_{0t} = 011)$ o $E_4(Q_{2t}\ Q_{1t}\ Q_{0t} = 100)$ según que la variable B posea el estado cero o el estado uno respectivamente. Esto se representa en las líneas cuarta y quinta de la tabla 6.24. La variable $\overline{C}/P$ debe adoptar el estado cero cuando el estado en el instante $t + 1$ es el siguiente al estado en el instante t, en binario o BCD natural. Por ello en la línea cuarta se asigna el estado cero a $\overline{C}/P$ y en la línea quinta, el estado uno. Se supone que el contador realiza el contaje en binario natural y por ello desde el estado E_7 se pasa al E_0 con $\overline{C}/P = 0$.

		t			$t+1$			
B	A	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	$\overline{C}/P$
X	X	0	0	0	0	0	1	0
X	0	0	0	1	0	1	0	0
X	1	0	0	1	1	1	0	1
0	X	0	1	0	0	1	1	0
1	X	0	1	0	1	0	0	1
X	X	0	1	1	1	0	1	1
X	X	1	0	0	1	0	1	0
X	X	1	0	1	0	0	0	1
X	X	1	1	0	1	1	1	0
X	X	1	1	1	0	0	0	0

TABLA 6.24

Partiendo de Q_{2t+1}, Q_{1t+1}, Q_{0t+1} y la variable $\overline{C}/P$ se obtiene el estado de las variables de entrada en paralelo D_2, D_1 y D_0. Cuando $\overline{C}/P$ adopta el estado lógico cero, es indiferente el estado de las variables D porque el contador pasará al estado siguiente al recibir un impulso en la entrada T. Cuando $\overline{C}/P$ adopta el estado lógico uno, el estado de las variables D debe ser idéntico al de las salidas Q del contador en el instante $t + 1$ (tabla 6.25).

Se pueden deducir las ecuaciones de las variables D_2, D_1 y D_0 y de $\overline{C}/P$ por el método de Karnaugh o el numérico. A continuación se utiliza el método de Karnaugh estudiado en el apartado 3.2.1 y se obtienen las tablas representadas en la figura 6.248 que se comentan seguidamente.

Para que en la tabla de D_2 no intervenga la variable B, se convierte en 1 la X de D_2 en la línea 4 de la tabla 6.25. Por la misma razón, la X de D_1 y D_0 de esa línea se convierte en 0.

		t						
B	A	Q_2	Q_1	Q_0	D_2	D_1	D_0	$\overline{C}/P$
X	X	0	0	0	X	X	X	0
X	0	0	0	1	X	X	X	0
X	1	0	0	1	1	1	0	1
0	X	0	1	0	X	X	X	0
1	X	0	1	0	1	0	0	1
X	X	0	1	1	1	0	1	1
X	X	1	0	0	X	X	X	0
X	X	1	0	1	0	0	0	1
X	X	1	1	0	X	X	X	0
X	X	1	1	1	X	X	X	0

TABLA 6.25

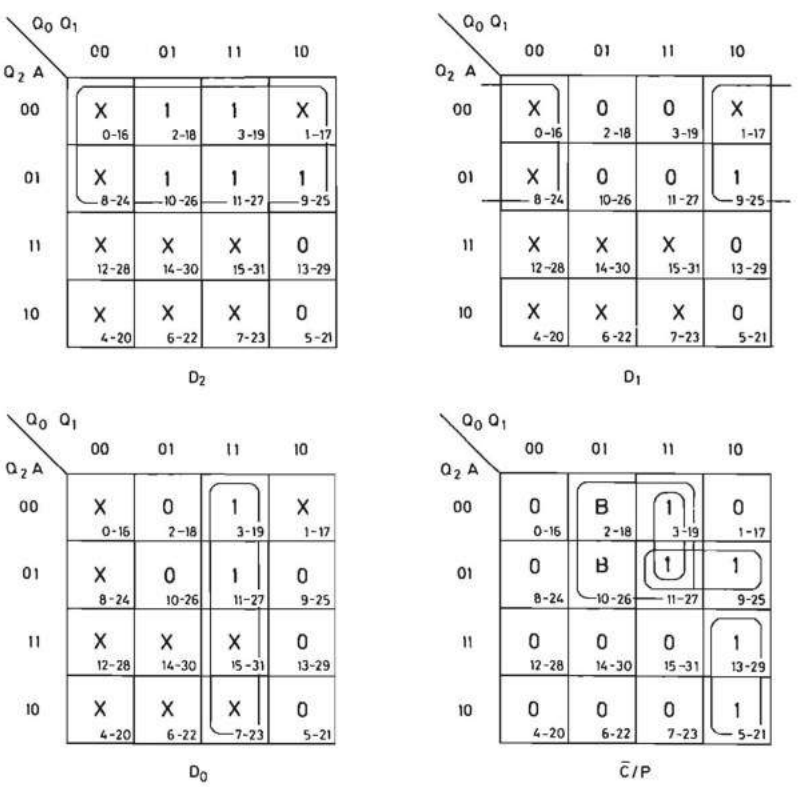


FIGURA 6.248.—**Tablas de Karnaugh del sistema secuencial síncrono realizado con un contador que ejecuta el diagrama de flujo de la figura 6.245.**

El valor de $\overline{C}/P$ cuando $Q_{2t}\ Q_{1t}\ Q_{0t} = 010$, coincide con B. Por ello en los cuadrados correspondientes de la tabla de Karnaugh de $\overline{C}/P$ se coloca la variable B.

De las tablas de la figura 6.248 resultan las ecuaciones:

$$D_2 = \overline{Q}_2$$
$$D_1 = \overline{Q}_1\ \overline{Q}_2$$
$$D_0 = Q_0\ Q_1$$
$$\overline{C}/P = Q_1\ \overline{Q}_2\ B + Q_0\ \overline{Q}_2\ A + Q_0\ \overline{Q}_1\ Q_2 + Q_0\ Q_1\ \overline{Q}_2$$

que convertidas adecuadamente resultan:

$$D_2 = \overline{Q}_2$$
$$D_1 = \overline{Q_1 + \overline{Q}_2}$$
$$D_0 = \overline{\overline{Q}_0 + \overline{Q}_1}$$
$$\overline{C}/P = \overline{Q_1\ \overline{Q}_2\ B}\ \ \overline{Q_0\ \overline{Q}_2\ A}\ \ \overline{Q_0\ \overline{Q}_1\ Q_2}\ \ \overline{Q_0\ Q_1\ \overline{Q}_2}$$

En la figura 6.249 se representa el esquema correspondiente. En el tomo II se estudian los procesadores digitales síncronos y, en particular, su síntesis con una unidad de control basada en el método que se acaba de desarrollar.

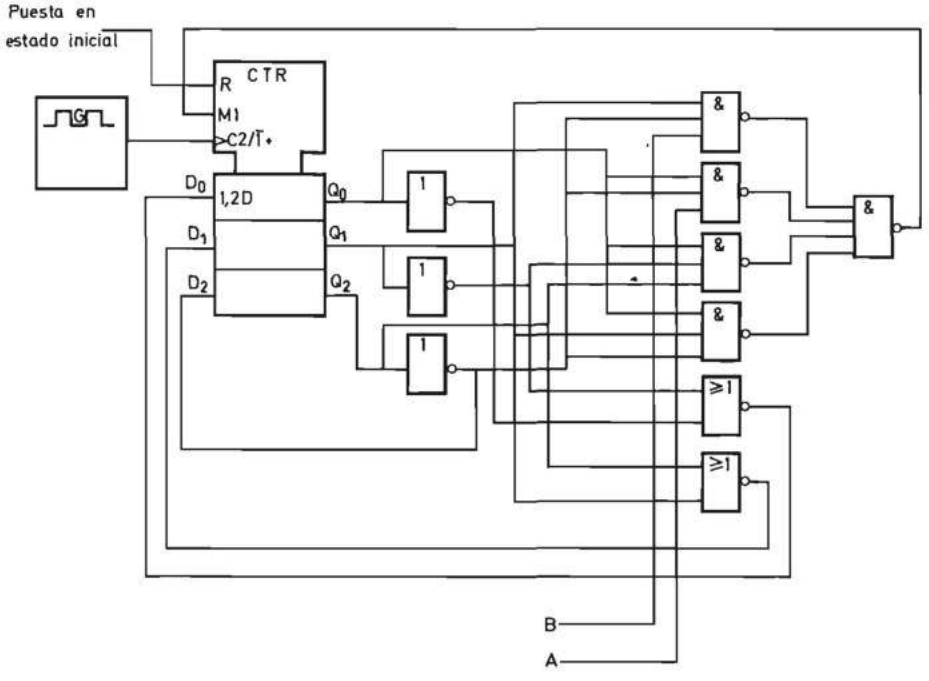


FIGURA 6.249.—**Esquema del sistema secuencial síncrono cableado realizado con un contador que ejecuta el diagrama de flujo de la figura 6.245.**

6.3.5.5.3 Síntesis de los sistemas secuenciales síncronos microprogramables. El esquema básico de un sistema secuencial síncrono microprogramable es el representado en la figura 6.250, que realiza la misma función que el sistema de la 6.155 con la única diferencia de que el sistema combinacional se especifica programable. Tal como se indicó anteriormente, la principal ventaja de la utilización de un sistema combinacional programable consiste en que se puede cambiar el diagrama de flujo del sistema secuencial síncrono sin modificar su cableado.

El sistema combinacional programable puede ser completo o incompleto. El registro síncrono se puede realizar también de diferentes maneras. Esto da lugar a

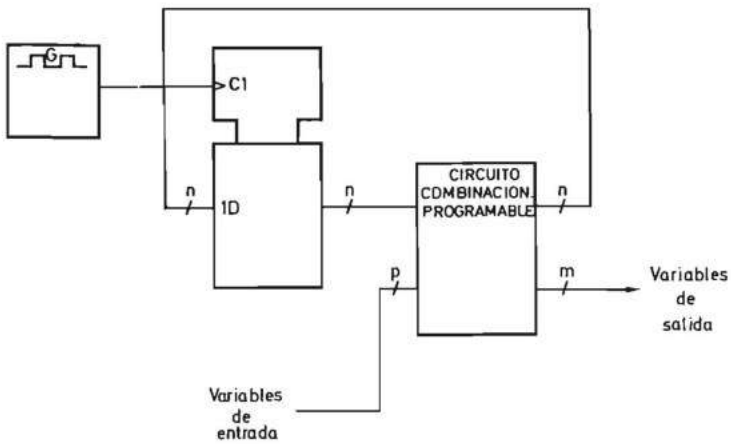


FIGURA 6.250.—**Esquema básico de un sistema secuencial síncrono microprogramable realizado con un registro.**

Sistema secuencial síncrono microprogramable	Sistema combinacional programable completo	Memoria de acceso aleatorio	+	Registro
		Memoria de acceso aleatorio	+	Contador síncrono con entradas en paralelo síncronas
	Sistema combinacional programable incompleto	Matriz lógica programable (PLA)	+	Registro (PLS)
		Matriz lógica programable (PLA)	+	Contador síncrono con entradas en paralelo síncronas
		Matriz lógica Y programable (PAL)	+	Registro
		Matriz lógica Y programable (PAL)	+	Contador síncrono con entradas en paralelo síncronas

TABLA 6.26

la clasificación de los sistemas secuenciales síncronos microprogramables indicada en la tabla 6.26.

En sucesivos apartados se estudian los diferentes tipos de sistemas secuenciales síncronos microprogramables y sus características.

6.3.5.5.3.1 Sistemas secuenciales síncronos microprogramables con circuitos combinacionales programables completos. El circuito combinacional completo más utilizado es la memoria de acceso aleatorio de escritura y lectura no simultáneas en cualquiera de sus versiones: activa (RAM), o pasiva (ROM, PROM, o RPROM).

En la figura 6.251 se representa el esquema general de un sistema secuencial síncrono microprogramable realizado con una memoria de acceso aleatorio. Este esquema constituye un caso particular de la figura 6.250.

El sistema secuencial evoluciona entre estados, a cada impulso de reloj, en función de las variables de entrada y del estado anterior. Cada uno de los estados viene determinado por el contenido del registro, y el comportamiento del sistema secuencial está caracterizado por sus variables de salida (variables de control).

En el esquema de la figura 6.251 el vector de salida de la memoria de acceso aleatorio consta de dos partes bien diferenciadas. Un vector de control (variables de salida del sistema secuencial) y un vector de estado (variables que determinan cuál es el próximo estado del sistema secuencial). En la figura 6.252 se representa de forma simbólica dicho vector de salida.

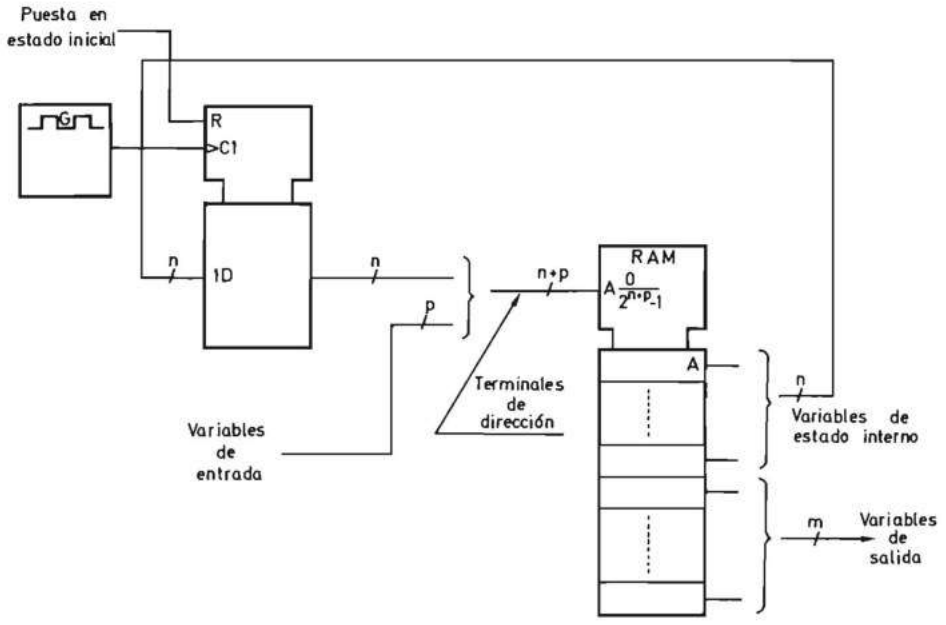


FIGURA 6.251.—**Esquema de un sistema secuencial síncrono microprogramable realizado con una memoria de acceso aleatorio y un registro.**

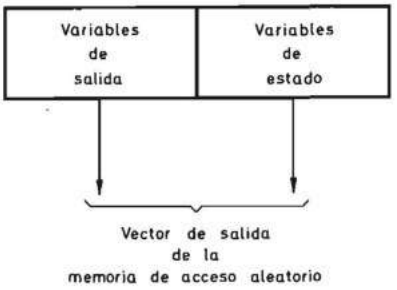


FIGURA 6.252.—**Vector de salida de la memoria de acceso aleatorio de la figura 6.251.**

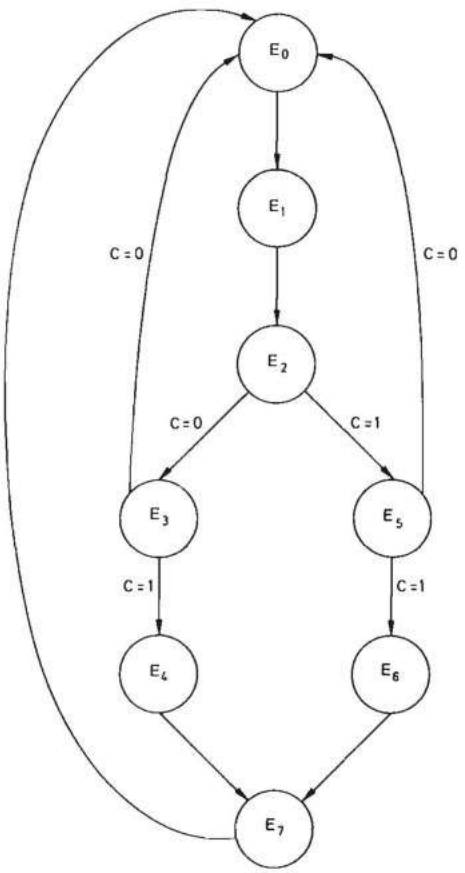


FIGURA 6.253.—**Ejemplo de diagrama de flujo de un sistema secuencial síncrono.**

El vector de estado interno y las variables de entrada constituyen la dirección de memoria que permitirá determinar el vector de salida en el próximo impulso de reloj.

Tal como se representó en la figura 6.244 el diagrama de flujo de un sistema secuencial síncrono está formado por un conjunto de estados internos entre los que se realizan transiciones incondicionales y condicionales en función del estado de una variable binaria, y puede ser realizado mediante el sistema de la figura 6.251.

Para demostrarlo se elige un diagrama de flujo adecuado que se representa en la figura 6.253.

Ejemplo 6.5:

Diseñar un sistema secuencial síncrono basado en el esquema de la figura 6.251, que evolucione de acuerdo con el diagrama de flujo de la figura 6.253.

A partir del diagrama de flujo se obtiene la tabla de verdad de la tabla 6.27.

En la figura 6.254 se representa el sistema diseñado formado por un registro síncrono de tres biestables activados por flancos (en la figura se suponen flancos de subida) y una memoria de acceso aleatorio.

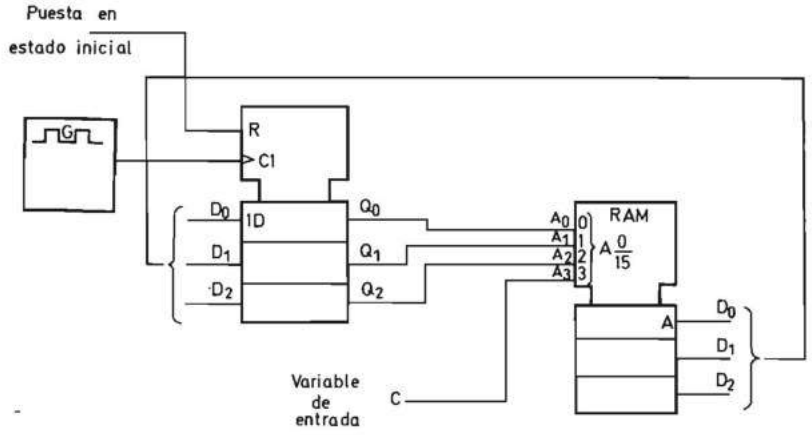


FIGURA 6.254.—**Esquema de un sistema secuencial síncrono microprogramable basado en el circuito de la figura 6.251 que ejecuta el diagrama de flujo de la figura 6.253.**

Las variables de salida del registro síncrono Q_2, Q_1, Q_0 (variables de estado) constituyen, en unión de la variable de entrada C, las variables de dirección de la memoria de acceso aleatorio. Arbitrariamente se decidió conectar Q_0, Q_1 y Q_2 a A_0, A_1 y A_2, respectivamente, y la variable C a A_3.

El contenido de la memoria de acceso aleatorio que hace evolucionar al sistema secuencial de acuerdo con el diagrama de estados anteriormente citado se muestra en la tabla 6.28. En cada posición de la memoria se coloca la información binaria equivalente al estado próximo indicado en la tabla 6.27. Por ejemplo, en la primera línea de la tabla 6.27 se indica que cuando el sistema se encuentra en el estado 000 debe pasar al 001 independientemente del valor de la variable C. Por ello en

Variable de entrada	Estado actual			Estado próximo		
C	Q_{2t}	Q_{1t}	Q_{0t}	Q_{2t+1}	Q_{1t+1}	Q_{0t+1}
X	0	0	0	0	0	1
X	0	0	1	0	1	0
0	0	1	0	0	1	1
1	0	1	0	1	0	1
0	0	1	1	0	0	0
1	0	1	1	1	0	0
X	1	0	0	1	1	1
0	1	0	1	0	0	0
1	1	0	1	1	1	0
X	1	1	0	1	1	1
X	1	1	1	0	0	0

TABLA 6.27

	Dirección				Contenido		
	A_3	A_2	A_1	A_0	D_2	D_1	D_0
0	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0
2	0	0	1	0	0	1	1
3	0	0	1	1	0	0	0
4	0	1	0	0	1	1	1
5	0	1	0	1	0	0	0
6	0	1	1	0	1	1	1
7	0	1	1	1	0	0	0
8	1	0	0	0	0	0	1
9	1	0	0	1	0	1	0
10	1	0	1	0	1	0	1
11	1	0	1	1	1	0	0
12	1	1	0	0	1	1	1
13	1	1	0	1	1	1	0
14	1	1	1	0	1	1	1
15	1	1	1	1	0	0	0

TABLA 6.28

la tabla 6.28 se coloca la información 001 en las posiciones de memoria 0000 (0_{10}) y 1000 (8_{10}).

Las variables D_0 a D_2 de salida de la memoria se conectan a las variables de entrada del registro síncrono.

Pero el diagrama de flujo de un sistema secuencial síncrono se puede realizar también con un sistema como el representado en la figura 6.255, que se obtiene a partir de la figura 6.251 añadiendo un circuito incrementador y un multiplexor.

El incrementador es un circuito combinacional que suma un uno binario a la combinación de entrada y presenta el valor incrementado a su salida. De esta forma es posible hacer que el sistema secuencial evolucione entre estados consecutivos o salte a otro estado (toma de decisión), dependiendo del valor de la variable de selección del multiplexor, que es un bit de cada una de las posiciones de memoria. Este sistema presenta la característica de que permite disponer simultáneamente de la información contenida en el registro y de la misma incrementada en una unidad, lo cual permite diseñar procesadores digitales secuenciales de gran velocidad tal como se estudia en el tomo II de este libro.

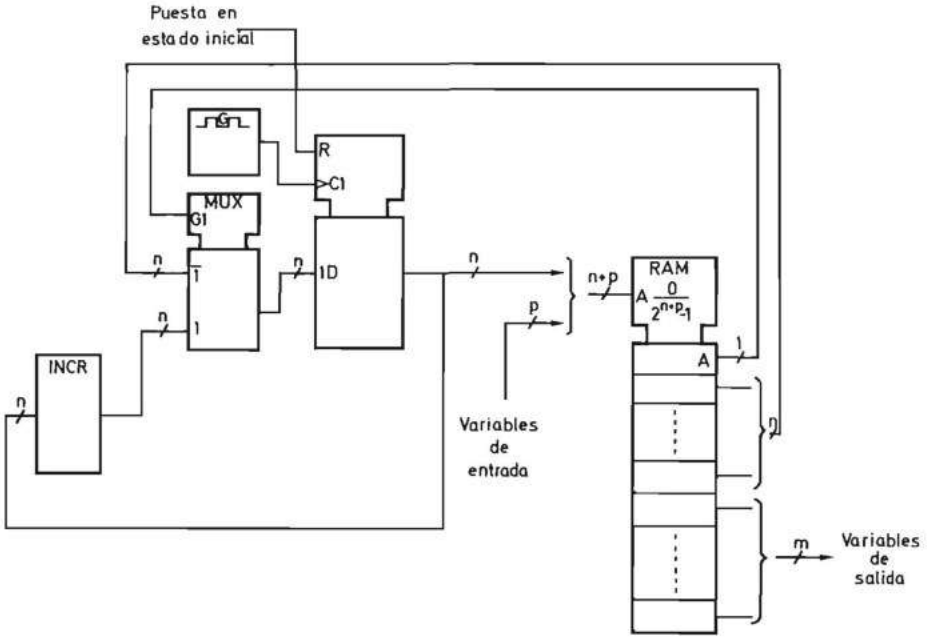


FIGURA 6.255.—**Esquema de un sistema secuencial síncrono microprogramable realizado con una memoria de acceso aleatorio, un registro, un incrementador y un multiplexor.**

Pero si se prescinde de la característica anteriormente citada se pueden sustituir los elementos incrementador, multiplexor y registro por un contador síncrono con entradas en paralelo síncronas, obteniéndose de esta forma el esquema de la figura 6.256. En las posiciones de la memoria de acceso aleatorio existe un bit que controla la variable $\overline{C}/P$ (contaje/paralelo) del contador y que según se encuentre en cero o en uno hace que el siguiente impulso del generador incremente en una unidad el contenido del contador o lo cambie por el campo de n bits de la posición de memoria que aparece en las entradas D en paralelo del contador.

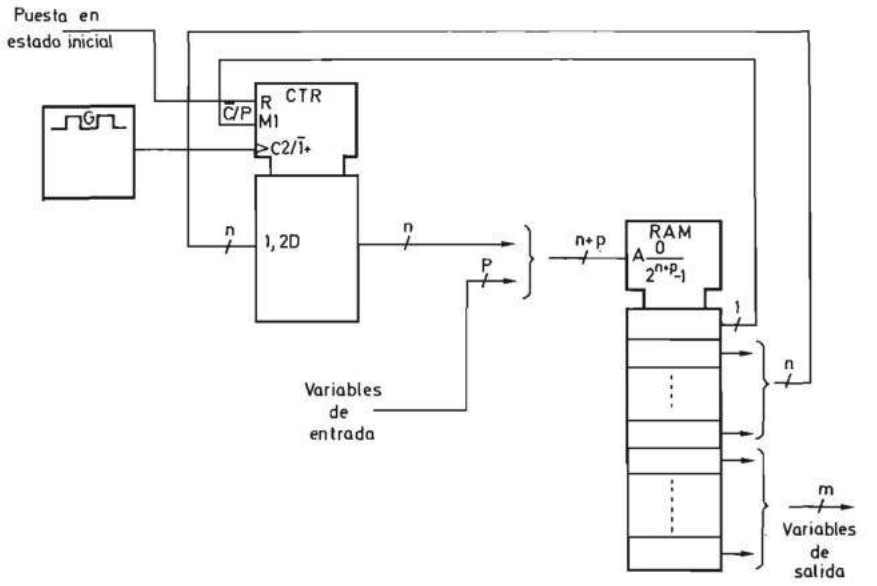


FIGURA 6.256.—**Esquema de un sistema secuencial síncrono microprogramable realizado con una memoria de acceso aleatorio y un contador.**

A continuación se realiza un ejemplo de diseño con un circuito basado en el esquema de la figura 6.256.

Ejemplo 6.6:

Diseñar, con un contador síncrono y una memoria de acceso aleatorio, un sistema secuencial síncrono que funcione de acuerdo con el diagrama de flujo de la figura 6.253.

El esquema correspondiente se representa en la figura 6.257. El funcionamiento del contador viene determinado por el valor lógico de la línea de control $\overline{C}/P$ (contaje/paralelo) de forma que, cuando el nivel de esta línea es bajo, el contador incrementa su contenido en una unidad en cada flanco procedente del generador de impulsos. Por el contrario, cuando su nivel es alto, los flancos aplicados a la entrada de sincronismo ($C2/\overline{1}+$) introducen en el contador la información presente en las entradas paralelo del mismo.

Por todo ello, en cada posición de memoria se ha de colocar un bit dedicado a la variable $\overline{C}/P$ además de los bits de las entradas en paralelo del contador.

En la tabla 6.29 se muestra el contenido que debe tener la memoria de acceso aleatorio para que el sistema secuencial evolucione de acuerdo con el diagrama de estados indicado en la figura 6.253. La variable D_3 contiene el estado de $\overline{C}/P$ y D_2, D_1, y D_0 las variables de estado interno. Cuando el sistema no tiene que saltar, D_3 se coloca en estado cero y es indiferente el estado de D_2, D_1 y D_0.

Este esquema solamente presenta la ventaja con respecto al de la figura 6.254 de que resulta más fácil la programación de la memoria, pero por el contrario ésta posee un bit más en cada una de sus posiciones.

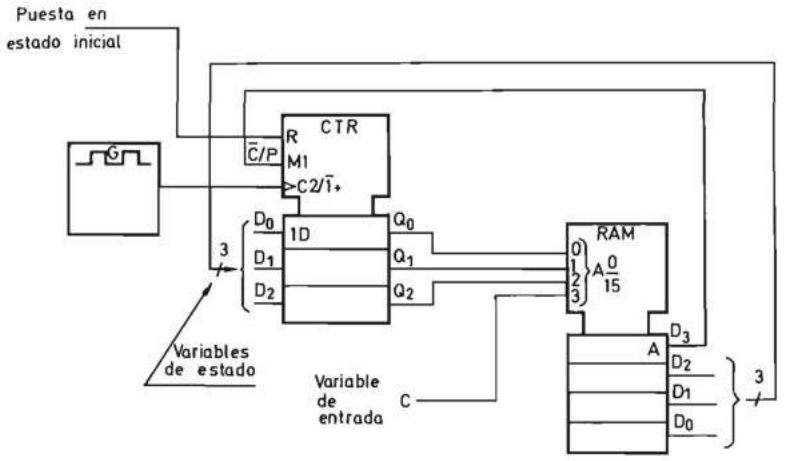


FIGURA 6.257—**Esquema de un sistema secuencial síncrono microprogramable basado en el circuito de la figura 6.256, que ejecuta el diagrama de flujo de la figura 6.253.**

Existen variantes de los circuitos de las figuras 6.251 y 6.256 que permiten reducir el número de líneas de dirección de la memoria de acceso aleatorio y, por lo tanto, su número de posiciones. Dichas variantes se representan en las figuras 6.258 y 6.259 en las que las p variables de entrada se conectan a las p entradas de un

	Dirección				Contenido			
	A_3	A_2	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	0	X	X	X
1	0	0	0	1	0	X	X	X
2	0	0	1	0	0	X	X	X
3	0	0	1	1	1	0	0	0
4	0	1	0	0	1	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	0	X	X	X
7	0	1	1	1	0	X	X	X
8	1	0	0	0	0	X	X	X
9	1	0	0	1	0	X	X	X
10	1	0	1	0	1	1	0	1
11	1	0	1	1	0	X	X	X
12	1	1	0	0	1	1	1	1
13	1	1	0	1	0	X	X	X
14	1	1	1	0	0	X	X	X
15	1	1	1	1	0	X	X	X

TABLA 6.29

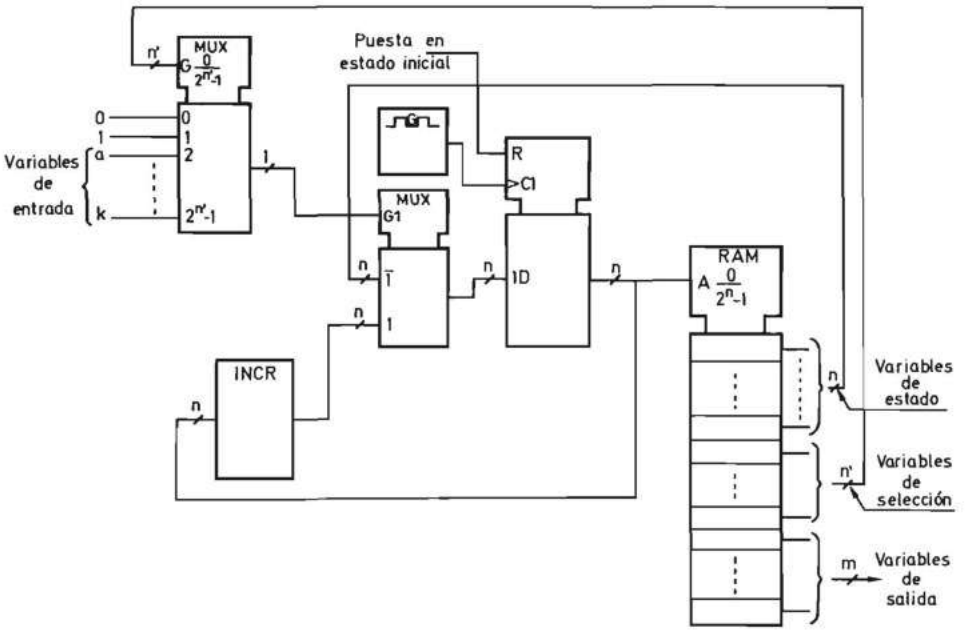


FIGURA 6.258.—**Esquema de un sistema secuencial síncrono microprogramable realizado con los mismos elementos que el de la figura 6.255, al que se le ha añadido un multiplexor de las variables de entrada.**

multiplexor que posee una variable de salida y n' variables de selección ($p<2^{n'}$), que son gobernadas por un campo de n' bits de las posiciones de la memoria. La combinación binaria de este campo selecciona una variable de entrada y, según que el estado de ésta sea un cero o un uno lógicos, el contenido del registro se incrementará en una unidad o será sustituido por el campo de dirección.

La salida de la memoria de acceso aleatorio puede, por lo tanto, ser considerada un vector con tres campos:

a) El campo de señales de control constituido por las variables de salida del sistema secuencial.

b) El campo de selección que permite determinar cuál de las variables de entrada estará presente a la salida del multiplexor.

c) El campo de dirección que es el nuevo estado al que irá el sistema secuencial si la toma de decisión resulta afirmativa. En caso contrario el próximo estado será consecutivo al anterior.

En la figura 6.260 se indica el formato de una microinstrucción de este tipo.

Los esquemas de las figuras 6.258 y 6.259 y los de las figuras 6.255 y 6.256 representan dos variantes extremas que admiten soluciones intermedias en que algunas variables de entrada se conectan a los terminales de dirección de la memoria de acceso aleatorio y otros a las entradas de un multiplexor.

A continuación se diseña un sistema secuencial síncrono de acuerdo con el esquema de la figura 6.259.

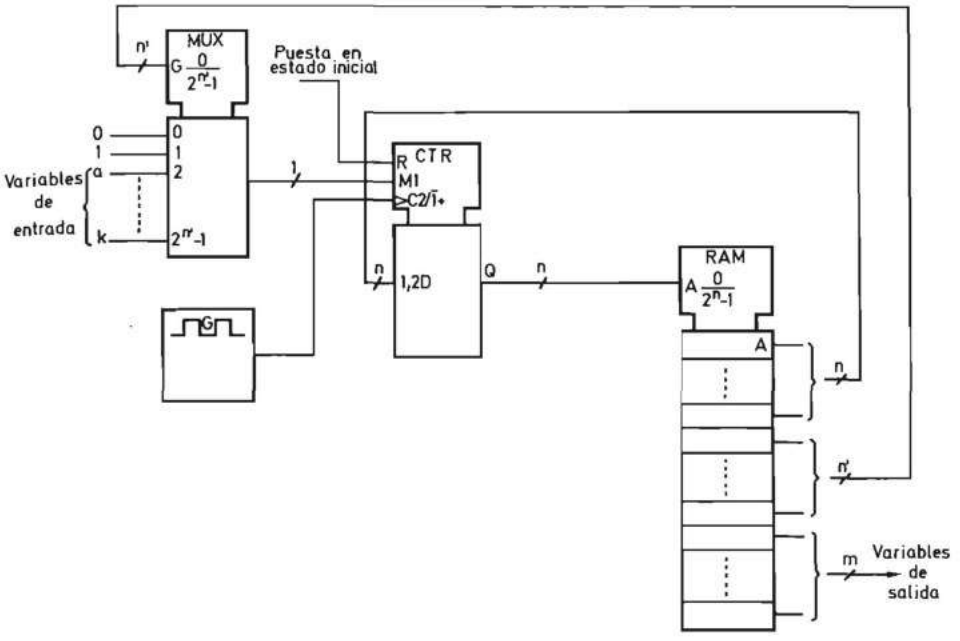


FIGURA 6.259.—**Esquema de un sistema secuencial síncrono microprogramable realizado con los mismos elementos que el de la figura 6.256, al que se ha añadido un multiplexor de las variables de entrada.**

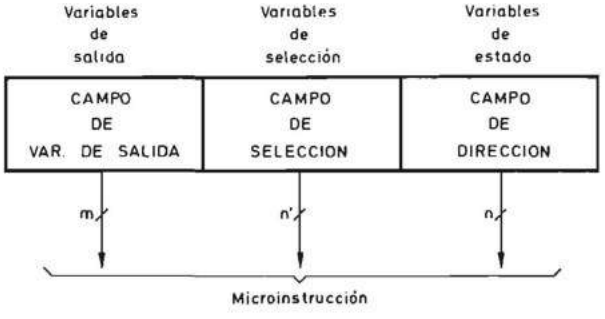


FIGURA 6.260.—**Formato de la microinstrucción de los sistemas secuenciales síncronos microprogramables de las figuras 6.258 y 6.259.**

Ejemplo 6.7:

Diseñar un sistema secuencial síncrono que funcione de acuerdo con el diagrama de flujo de la figura 6.253, utilizando para ello un contador síncrono, una memoria de acceso aleatorio y un multiplexor de las variables de entrada.

En la figura 6.261 se muestra el sistema obtenido. En este caso el multiplexor ha de tener cuatro entradas que permitan seleccionar las cuatro condiciones de evolu-

ción: incremento del contador, salto incondicional a un nuevo estado, salto condicional en función de la variable C y salto condicional en función de la variable $\overline{C}$. Se utiliza para ello un multiplexor de cuatro canales.

En cada posición de memoria existe una información dividida en dos campos: un campo de selección de dos bits que controla las entradas de selección S_1 y S_0 del multiplexor y un campo que contiene las variables de estado que se conectan a las entradas en paralelo del contador. En la tabla 6.30 se indica el funcionamiento del sistema secuencial dependiendo del valor de las variables de selección. La memoria de acceso aleatorio posee menos posiciones que las de las figuras 6.254 y 6.257 porque su número de posiciones es igual al de estados internos que tiene el sistema, dado que las variables de entrada se conectan al multiplexor en lugar de a la memoria.

D_4 \ D_3	0	1
0	Incrementa	Salto incondicional
1	Salto condicional en función de la variable $\overline{C}$	Salto condicional en función de la variable C

TABLA 6.30

	Dirección			Contenido				
	A_2	A_1	A_0	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	X	X	X
1	0	0	1	0	0	X	X	X
2	0	1	0	1	1	1	0	1
3	0	1	1	1	0	0	0	0
4	1	0	0	0	1	1	1	1
5	1	0	1	1	0	0	0	0
6	1	1	0	0	0	X	X	X
7	1	1	1	0	0	X	X	X

TABLA 6.31

En la tabla 6.31 se muestra el contenido que debe poseer la memoria de acceso aleatorio para que el sistema secuencial de la figura 6.261 evolucione de acuerdo con el diagrama de estados de la figura 6.253.

La memoria posee ocho posiciones correspondientes a los ocho estados internos. Como ejemplo, en la cuarta fila correspondiente al estado 3(011) los bits S_i

y S_0 se colocan en los niveles 1 y 0 respectivamente, con lo cual en la salida del multiplexor aparece el valor de la variable $\overline{C}$. En los otros tres bits se pone la información 000 para que si $\overline{C}$ tiene el valor lógico uno se produzca el salto al estado 0(000).

Como resumen de todo lo expuesto se debe resaltar que tanto en el esquema de la figura 6.257 como en el de la 6.261 existen muchos bits en las memorias de acceso aleatorio cuyo estado es indiferente (tablas 6.29 y 6.31).

Esto lleva a pensar en las ventajas de la utilización de un sistema combinacional programable incompleto como indicaremos en el apartado siguiente.

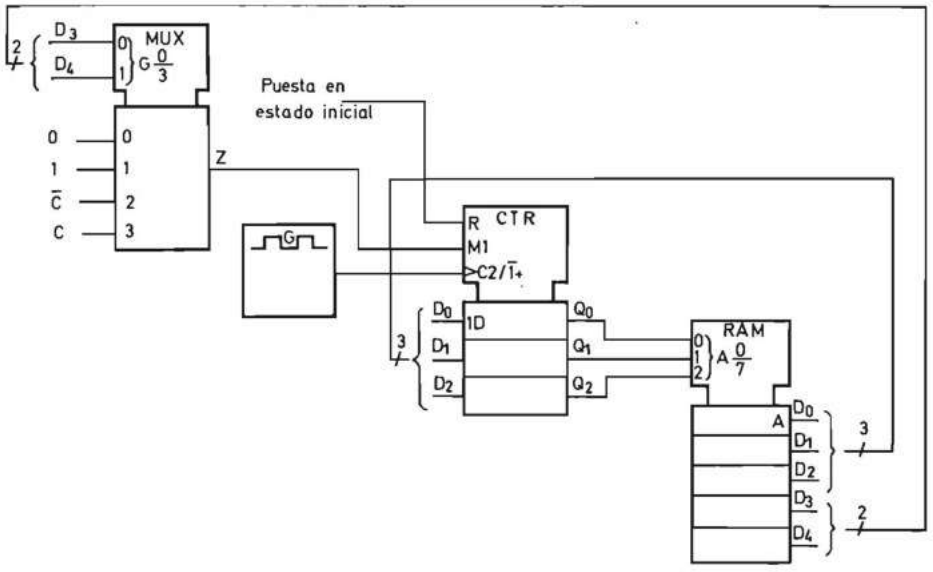


FIGURA 6.261.—**Esquema de un sistema secuencial síncrono microprogramable basado en el circuito de la figura 6.259 que ejecuta el diagrama de flujo de la figura 6.253.**

Existe, por otra parte, una alternativa que permite la toma de decisiones múltiples. En la figura 6.262 se representa un sistema secuencial síncrono que puede tomar la decisión de saltar a tres estados distintos. Para ello la memoria de acceso aleatorio tiene tres campos de dirección que se conectan a otras tantas entradas de un multiplexor que se seleccionan mediante las salidas de dos multiplexores controlados mediante el campo de selección. En la figura 6.263 se representa el vector de salida de la memoria de acceso aleatorio que constituye la microinstrucción. La complejidad adicional en la longitud de la posición de la memoria de acceso aleatorio ha hecho que esta solución sea poco utilizada en la práctica.

6.3.5.5.3.2 Sistemas secuenciales síncronos microprogramables con circuitos combinacionales programables incompletos. En el apartado anterior se ha estudiado la realización de sistemas secuenciales microprogramables con circuitos com-

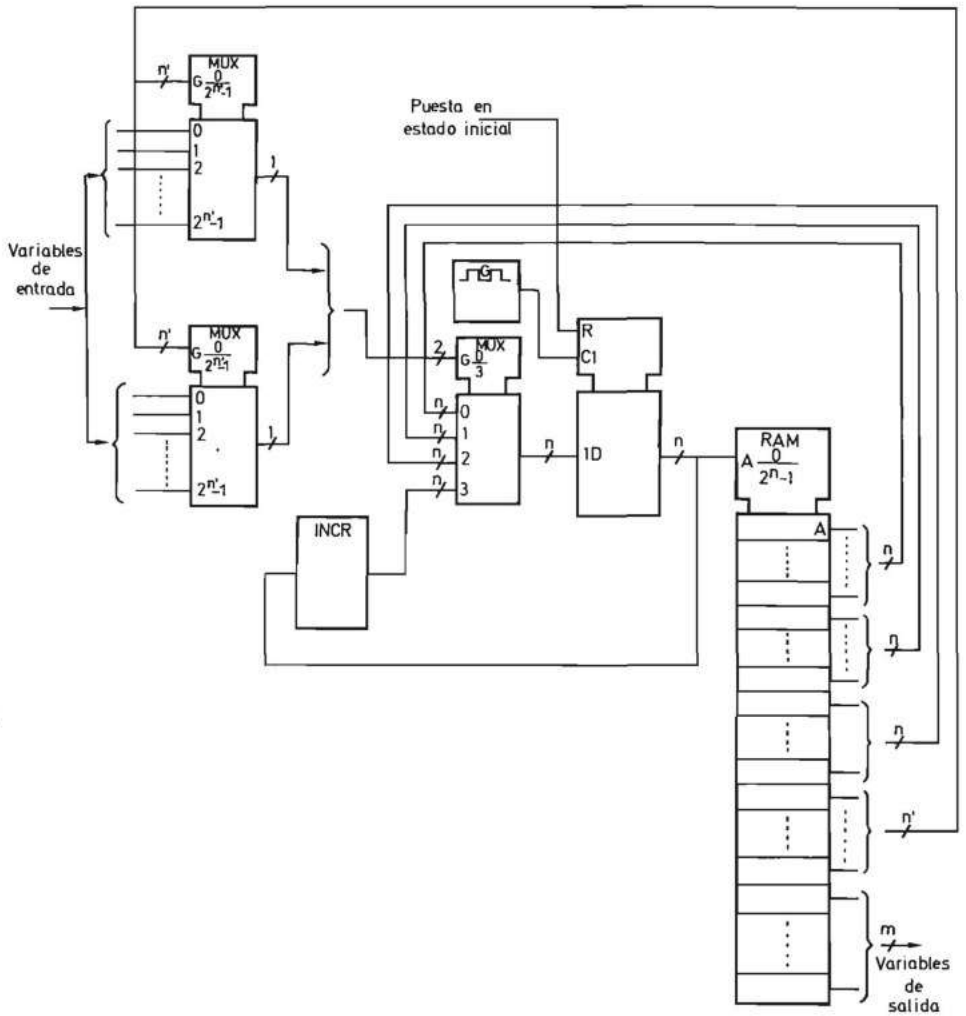


FIGURA 6.262.—**Esquema de un sistema secuencial síncrono microprogramable con tomas de decisión múltiples.**

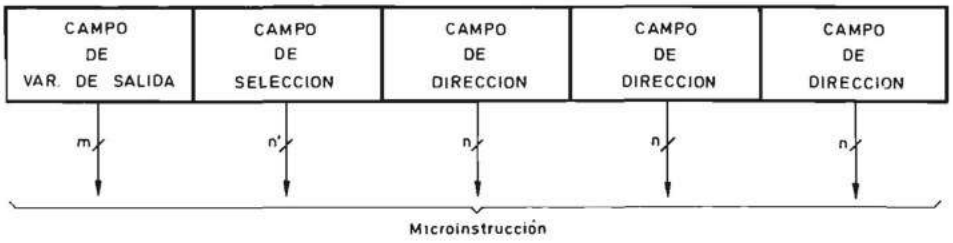


FIGURA 6.263.—**Formato de la microinstrucción del sistema secuencial síncrono microprogramable de la figura 6.262.**

binacionales programables completos, en particular con memorias de acceso aleatorio.

Analizando con mayor detenimiento las soluciones propuestas en el apartado anterior, se llega a la conclusión de que los circuitos combinacionales completos presentan el inconveniente de que cuando la variable o variables de selección (figs. 6.256 y 6.259) adoptan un nivel lógico tal que el contador incrementa su contenido, es indiferente el estado de las variables de dirección. Se puede, por lo tanto, afirmar que la memoria de acceso aleatorio posee un número de bits mayor del necesario, con el correspondiente aumento de complejidad y de disipación.

Este inconveniente se elimina sustituyendo el circuito combinacional programable completo por uno incompleto.

En la figura 6.264 se representa el esquema general de un sistema secuencial síncrono realizado con un circuito combinacional programable (CCP) incompleto.

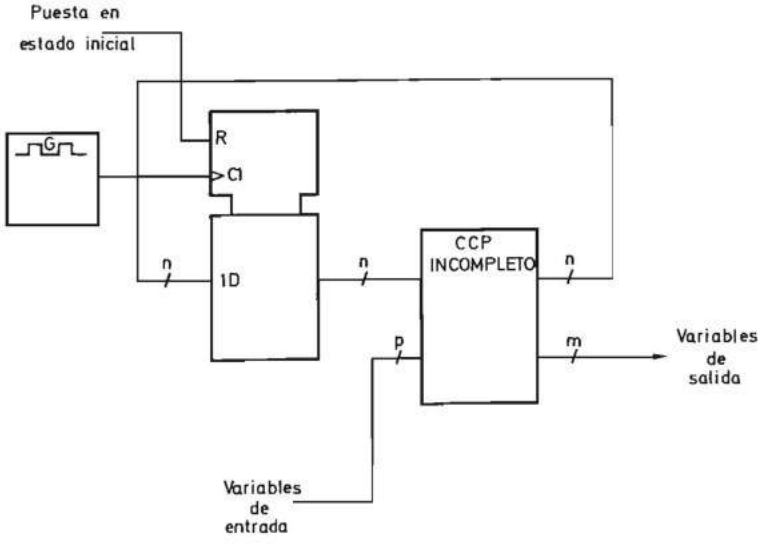


FIGURA 6.264.—**Esquema general de un sistema secuencial síncrono microprogramable realizado con un registro y un circuito combinacional programable (CCP) incompleto.**

Cualquiera de los esquemas desarrollados en el apartado anterior con CCP completos (figuras 6.254, 6.257 y 6.261) puede ser realizado con uno incompleto que resulta más sencillo.

Por otra parte, debido a que el circuito de la figura 6.264 es de aplicación general, ha sido realizado en un circuito integrado bajo el nombre de secuenciador lógico programable [Programmable logic sequencer (PLS)] [PHIL 87].

El CCP incompleto puede ser una matriz lógica programable (PLA) o una matriz lógica Y-programable (PAL). A continuación se diseña un sistema secuencial síncrono con ambas para comprobar las ventajas que presenta su utilización con respecto a las memorias de acceso aleatorio.

Ejemplo 6.8:

Diseñar un sistema secuencial síncrono que evolucione de acuerdo con el diagrama de flujo de la figura 6.253, utilizando para ello un circuito combinacional programable.

El primero se realiza con una matriz lógica programable y un registro síncrono, obteniéndose el esquema de la figura 6.265*a*.

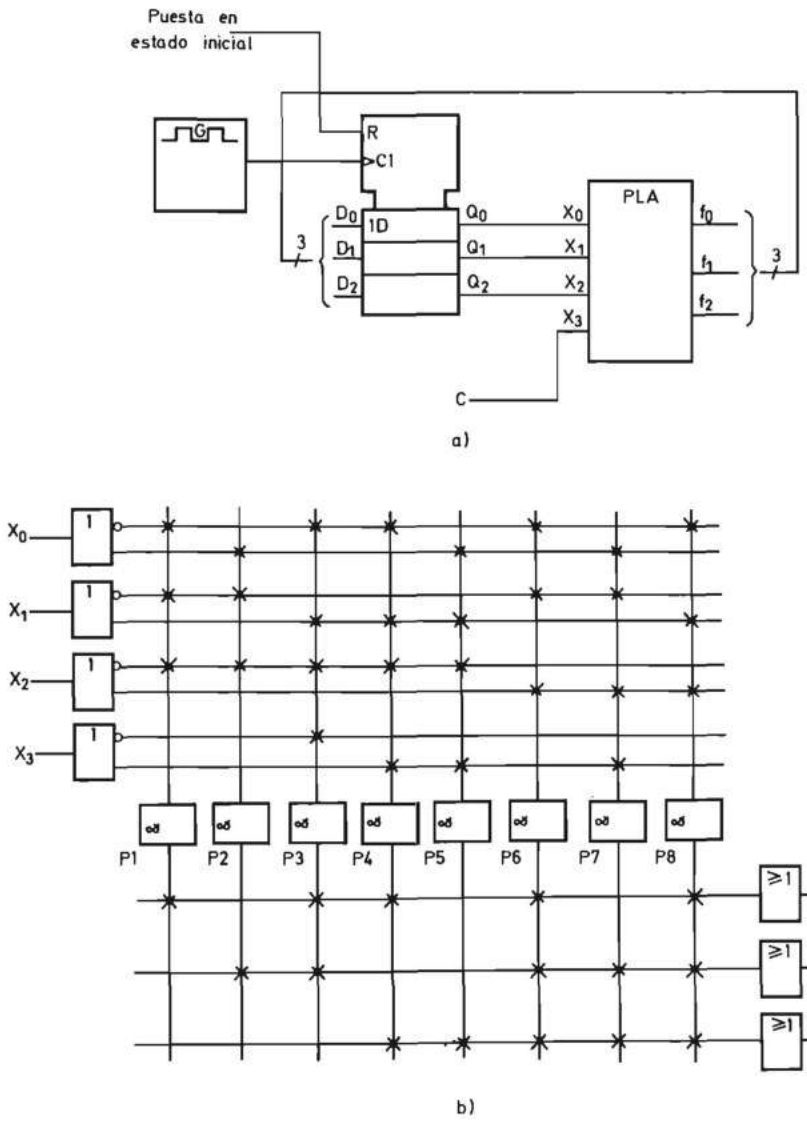


FIGURA 6.265.—**Sistema secuencial síncrono microprogramable realizado con un registro y una matriz lógica programable (PLA): *a*) Esquema del circuito; *b*) Programación de la matriz lógica programable (PLA) para que ejecute el diagrama de flujo de la figura 6.253.**

Producto n.°	Entradas X_3	X_2	X_1	X_0	Salidas f_2	f_1	f_0
1	X	0	0	0	0	0	1
2	X	0	0	1	0	1	0
3	0	0	1	0	0	1	1
4	1	0	1	0	1	0	1
	0	0	1	1	0	0	0
5	1	0	1	1	1	0	0
6	X	1	0	0	1	1	1
	0	1	0	1	0	0	0
7	1	1	0	1	1	1	0
8	X	1	1	0	1	1	1
	X	1	1	1	0	0	0

TABLA 6.32

En la tabla 6.32 se indican los valores de las variables de salida correspondientes a cada una de las combinaciones posibles de las variables de entrada para que el sistema secuencial de la figura 6.265 evolucione de acuerdo con el diagrama de estados de la figura 6.253.

En el caso de que no se minimicen las funciones, la matriz lógica programable ha de tener un número de puertas Y igual al de filas de la tabla 6.32 que poseen un uno lógico en al menos una de las funciones f_0, f_1 o f_2. En el caso que estamos estudiando, el número de puertas Y necesario es de 8, las cuales se numeran de forma correlativa (tabla 6.32).

Producto n.°	Entradas X_3	X_2	X_1	X_0	Salidas f_3	f_2	f_1	f_0
	X	0	0	0	0	X	X	X
	X	0	0	1	0	X	X	X
	0	0	1	0	0	X	X	X
1	1	0	1	0	1	1	0	1
2	0	0	1	1	1	0	0	0
	1	0	1	1	0	X	X	X
3	X	1	0	0	1	1	1	1
4	0	1	0	1	1	0	0	0
	1	1	0	1	0	X	X	X
	X	1	1	0	0	X	X	X
	X	1	1	1	0	X	X	X

TABLA 6.33

En la figura 6.265*b* se representa la matriz lógica programable. Para simplificar el esquema se indica una sola línea de entrada a cada puerta Y y cada puerta O. En los puntos en que se coloca una cruz en la matriz de puertas Y, quiere decir que la variable de entrada correspondiente a la columna se conecta a una línea de entrada de la puerta Y. Lo mismo se puede decir de la matriz de puertas O. Por ejemplo, para formar el producto 8 no se conecta la variable X_3 porque las salidas son independientes de su valor. Por el contrario, se coloca una cruz en las verticales correspondientes a X_2, X_1 y $\overline{X}_0$, porque las dos primeras toman el valor uno y la última el valor cero. En la matriz de puertas O se procede de una forma similar. Por ejemplo, en la vertical correspondiente a f_0 se colocan cruces en las líneas 1, 3, 4, 6 y 8 porque para dichos productos la función f_0 forman el valor uno, como puede verse en la tabla 6.32.

El resultado alcanzado es una reducción de complejidad apreciable en la realización del circuito combinacional programable, ya que la matriz lógica programable tiene sólo ocho términos producto frente a los dieciséis que presenta la memoria de acceso aleatorio para la realización del sistema secuencial particular que estamos estudiando.

En la figura 6.266*a* se representa el esquema de la realización con una matriz lógica Y-programable (PAL) que coincide con el de la figura 6.266*b* sin más que sustituir las siglas PAL por PLA. El hecho de no ser programable la matriz de puertas O, hace que (en el supuesto de que cada puerta O esté conectada al mismo número de puertas Y) el número mínimo de puertas Y sea igual al de funciones multiplicado por el de unos de la función que posea mayor número de ellos en la tabla 6.32. Dado que en dicha tabla las tres funciones poseen cinco unos, la PAL debe tener 15 puertas Y, tal como se indica en la figura 6.266*b*, en la que se observa que las puertas Y P_2 y P_7 realizan el mismo producto, al igual que P_4, P_8 y P_{13}.

Es interesante resaltar que la PAL de la figura 6.266*b* posee 7 puertas Y más que la PLA de la figura 6.266*a* pero, en contrapartida, no es necesario en ella programar la matriz de puertas O.

Minimizando las funciones f_0, f_1 y f_2 por Karnaugh u otro método, se puede simplificar la PLA o la PAL utilizadas tal como se vio en el apartado 3.8.2. Esto es interesante en los siguientes casos:

a) Cuando el sistema secuencial síncrono forme parte de un circuito más complejo integrado (LSI o VLSI) de aplicación general o a medida (custom design).
b) Cuando la simplificación permita utilizar un bloque funcional matriz lógica programable (PLA) o matriz lógica Y-programable (PAL) más sencillos.

Se puede realizar también este sistema secuencial síncrono con un registro, un incrementador y un multiplexor combinados con un PLA o una PAL. Igualmente se puede realizar con un contador con entradas en paralelo síncronas y una PLA o una PAL. En ambos casos se obtiene, en general, una reducción de la complejidad de la PLA pero no así de la PAL.

En la figura 6.267*a* se representa un sistema secuencial síncrono realizado con un contador cuyo comportamiento puede ser idéntico al del sistema de la figura 6.265*a*. En la tabla 6.33 se indican los valores de las variables de salida de la matriz correspondientes a las distintas combinaciones de entrada para que el sistema se-

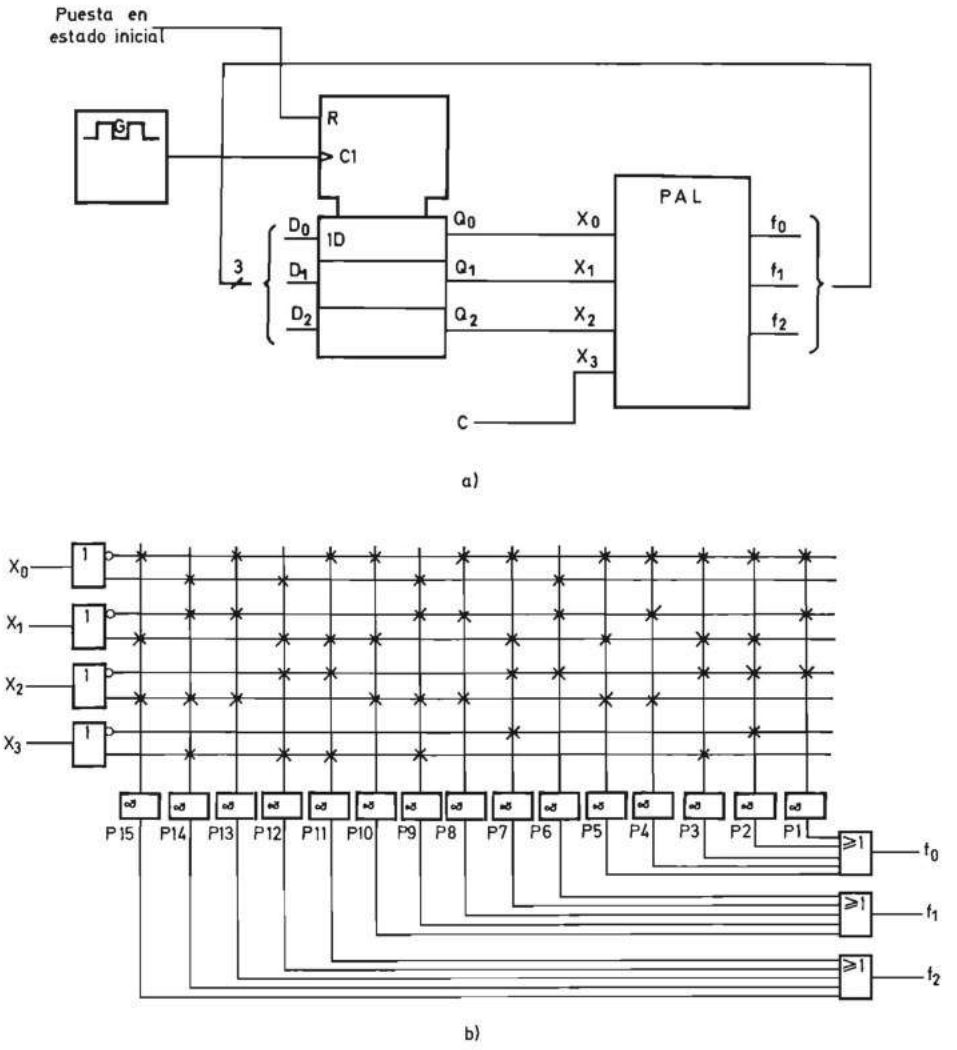


FIGURA 6.266.—**Sistema secuencial síncrono microprogramable realizado con un registro y una matriz lógica Y-programable (PAL): *a*) Esquema del circuito; *b*) Programación de la matriz lógica Y-programable (PAL) para que ejecute el diagrama de flujo de la figura 6.253.**

cuencial de la figura 6.267*a* evolucione de acuerdo con el diagrama de estados de la figura 6.253.

Al igual que en el apartado anterior, en el caso de utilizar una matriz lógica programable, sólo es necesario programar los productos de las variables de entrada correspondientes a aquellas líneas de la tabla 6.33 en que alguna salida adopta el nivel uno. Resultan, por lo tanto, en este caso cuatro productos y la matriz lógica programable obtenida se representa en la figura 6.267*b*.

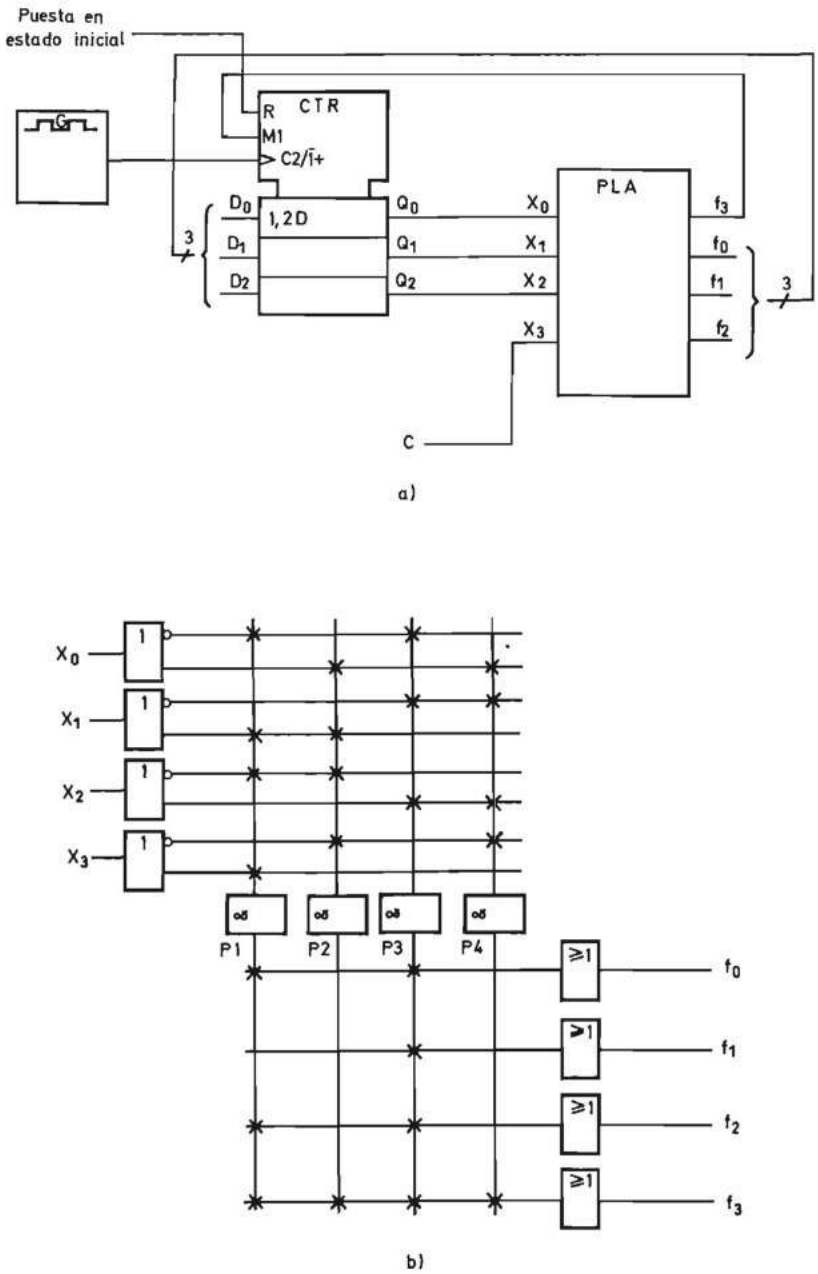


FIGURA 6.267.—**Sistema secuencial síncrono microprogramable realizado con un contador y una matriz lógica programable (PLA): *a*) Esquema del circuito; *b*) Programación de la matriz lógica programable (PLA) para que ejecute el diagrama de flujo de la figura 6.253.**

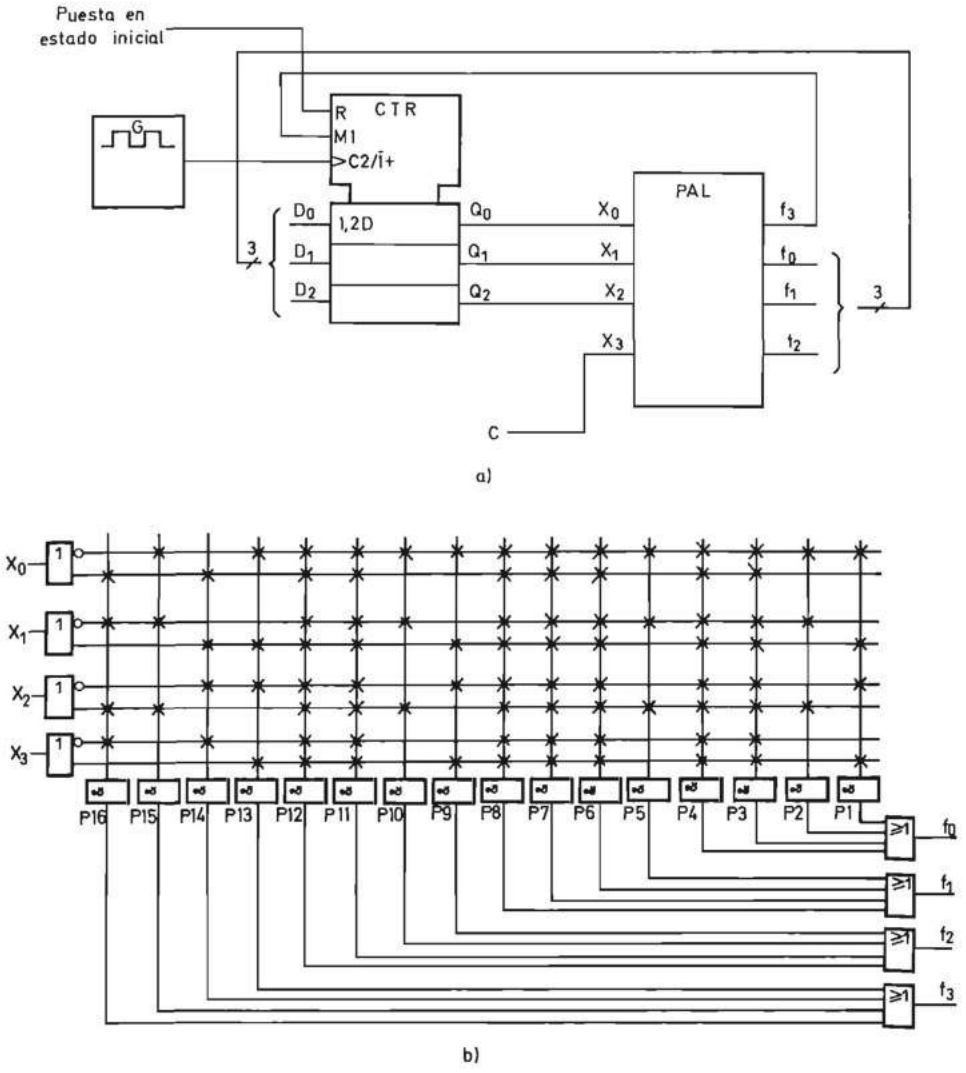


FIGURA 6.268.—**Sistema secuencial síncrono microprogramable realizado con un contador y una matriz lógica Y-programable (PAL): *a*) Esquema del circuito; *b*) Programación de la matriz lógica Y-programable (PAL) para que ejecute el diagrama de flujo de la figura 6.254.**

Si se utiliza una matriz lógica Y-programable es necesario que ésta tenga como mínimo cuatro grupos de cuatro puertas Y cada uno, tal como se representa en la figura 6.268*b* en la que se observa que en las funciones f_0 y f_2 sólo se utilizan dos productos y en f_1 sólo se utiliza uno. En las puertas Y que no se utilizan no se suprime ninguna conexión para que su salida se encuentre siempre a nivel cero.

El lector también puede observar que algunos de los productos están repetidos (por ejemplo, P_2 y P_5 realizan el mismo producto). La consecuencia de todo ello

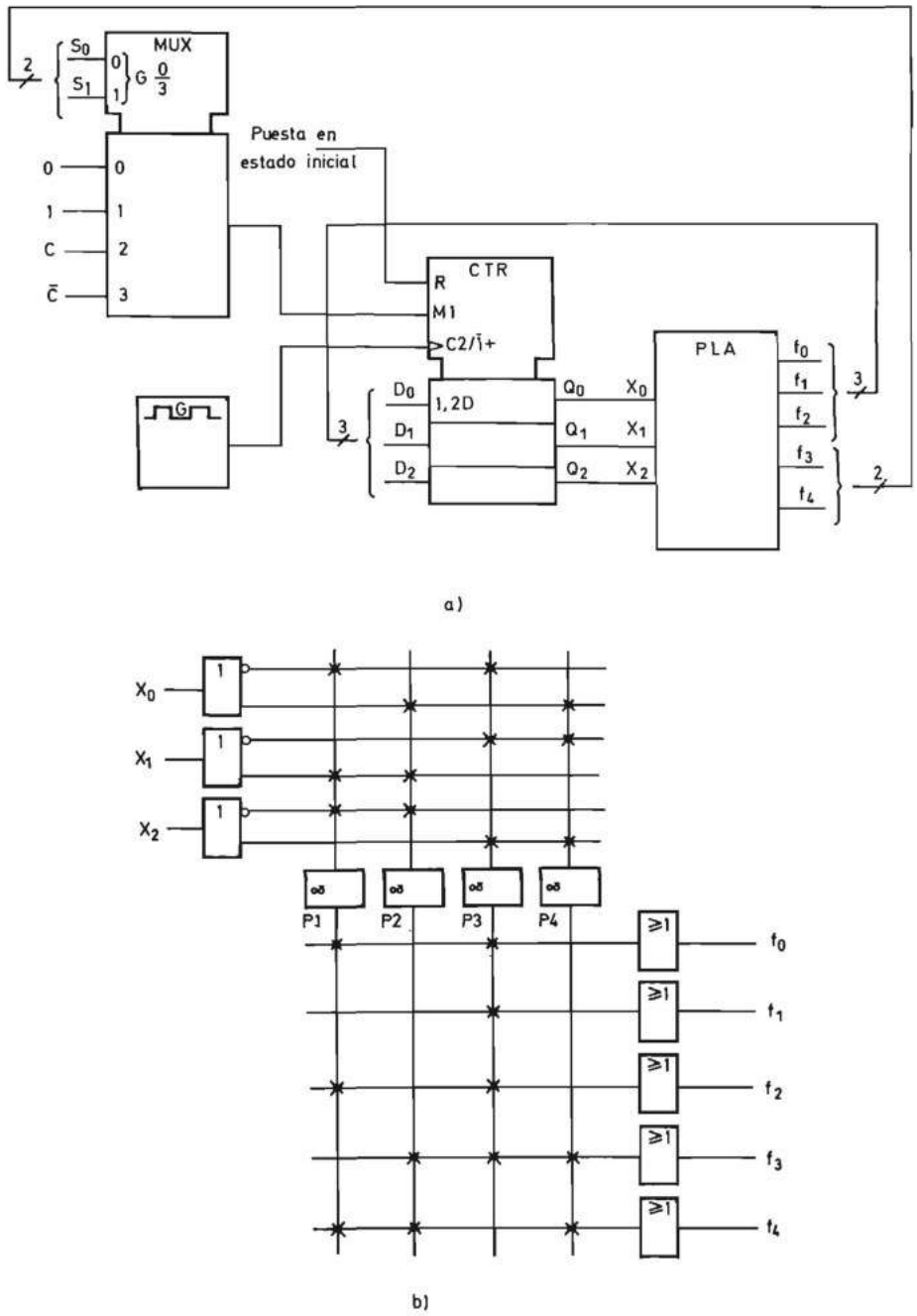
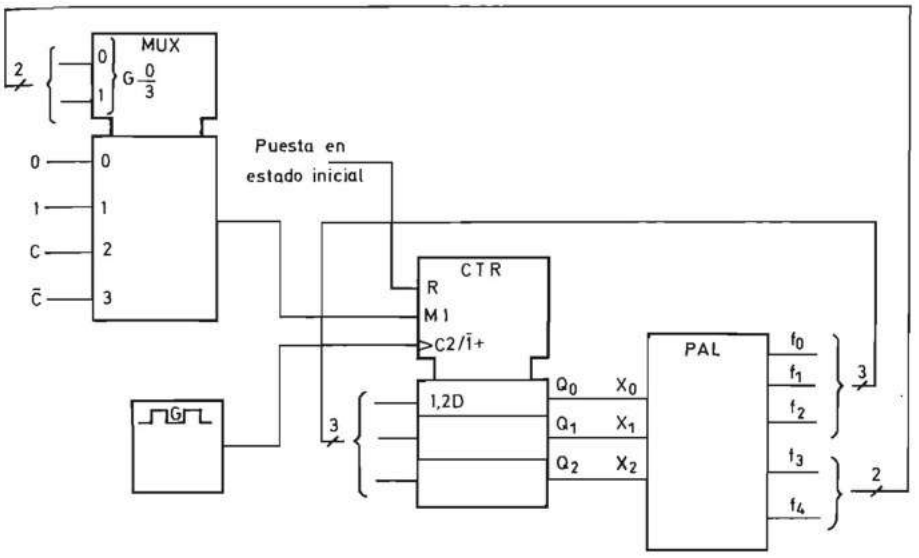


FIGURA 6.269.—**Sistema secuencial síncrono microprogramable realizado con un contador, una matriz lógica programable (PLA) y un multiplexor de las variables de entrada: *a*) Esquema del circuito; *b*) Programación de la matriz lógica programable (PLA) para que ejecute el diagrama de flujo de la figura 6.253.**



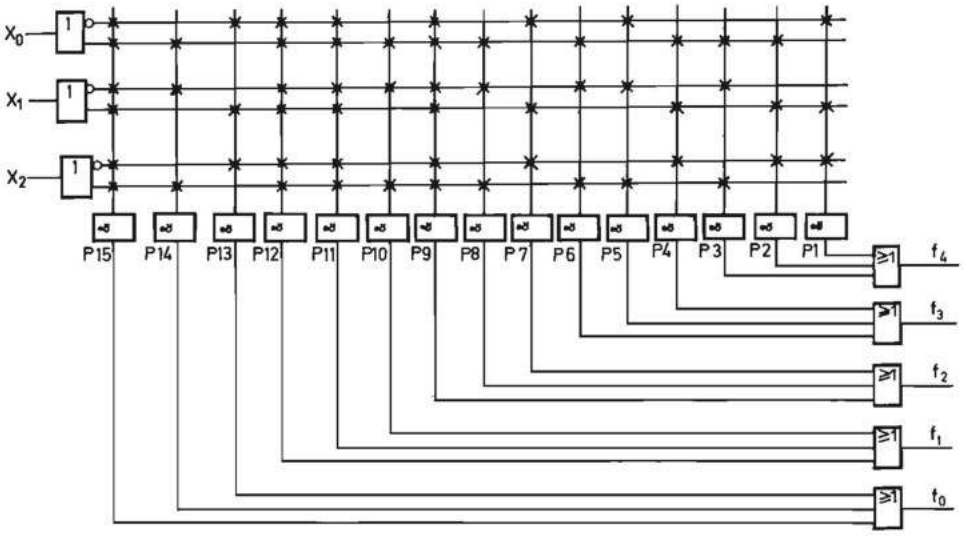


FIGURA 6.270.—**Sistema secuencial síncrono microprogramable realizado con un contador, una matriz lógica Y-programable (PAL) y un multiplexor de las variables de entrada: *a*) Esquema del circuito; *b*) Programación de la matriz lógica Y-programable para que ejecute el diagrama de flujo de la figura 6.253.**

es que esta PAL resulta más complicada que la de la figura 6.266 y, por lo tanto, la utilización de un contador no presenta ventajas con respecto a un registro.

La PLA o la PAL se pueden combinar también con un multiplexor obteniéndose los esquemas de las figuras 6.269*a* y 6.270*a*.

En la tabla 6.34 se muestra la tabla de verdad que debe realizar el circuito combinacional.

Si se realiza con una PLA es necesario que ésta posea cuatro productos, tal como se indica en la figura 6.269*b*. Dichos productos corresponden a las líneas 1, 2, 3 y 4 de la tabla 6.34. El lector puede comprobar si la minimización de las funciones f_0 a f_4 permite o no reducir el tamaño de la PLA.

Producto n.º	Entradas			Salidas				
	X_2	X_1	X_0	S_1 f_4	S_0 f_3	D_2 f_2	D_1 f_1	D_0 f_0
	0	0	0	0	0	X	X	X
	0	0	1	0	0	X	X	X
1	0	1	0	1	0	1	0	1
2	0	1	1	1	1	0	0	0
3	1	0	0	0	1	1	1	1
4	1	0	1	1	1	0	0	0
	1	1	0	0	0	X	X	X
	1	1	1	0	0	X	X	X

TABLA 6.34

En la figura 6.270*b* se representa la PAL mínima necesaria que posee 15 productos de los que sólo se utilizan once.

Este resultado confirma que los sistemas secuenciales síncronos realizados con contadores en combinación con PAL no son más sencillos que los que utilizan registros. Por ello, los circuitos disponibles en gran escala de integración que reciben el nombre de dispositivos lógicos programables (Programmable Logic Devices) utilizan matrices lógicas Y-programables (PAL) y registros [INTE 88] [MAND 90].

6.3.5.5.4 Sistemas secuenciales síncronos modulares. En el apartado 6.3.5.5.3 se estudian los sistemas secuenciales síncronos microprogramables y en él se analizan diversas formas de realizarlos, entre las que se encuentran las representadas en las figuras 6.258 y 6.259 en las que las variables de entrada se aplican al sistema a través de un multiplexor. En la figura 6.271*a* se representa un sistema secuencial síncrono, obtenido partir del de la figura 6.259, en el cual se dedica un multiplexor MUX1 a las variables de entrada y se coloca entre éste y el contador un segundo multiplexor MUX2 de cuatro canales conectados respectivamente a cero, uno, la salida directa del MUX1 (Z) y la salida inversa ($\overline{Z}$). En la figura 6.271*b* se

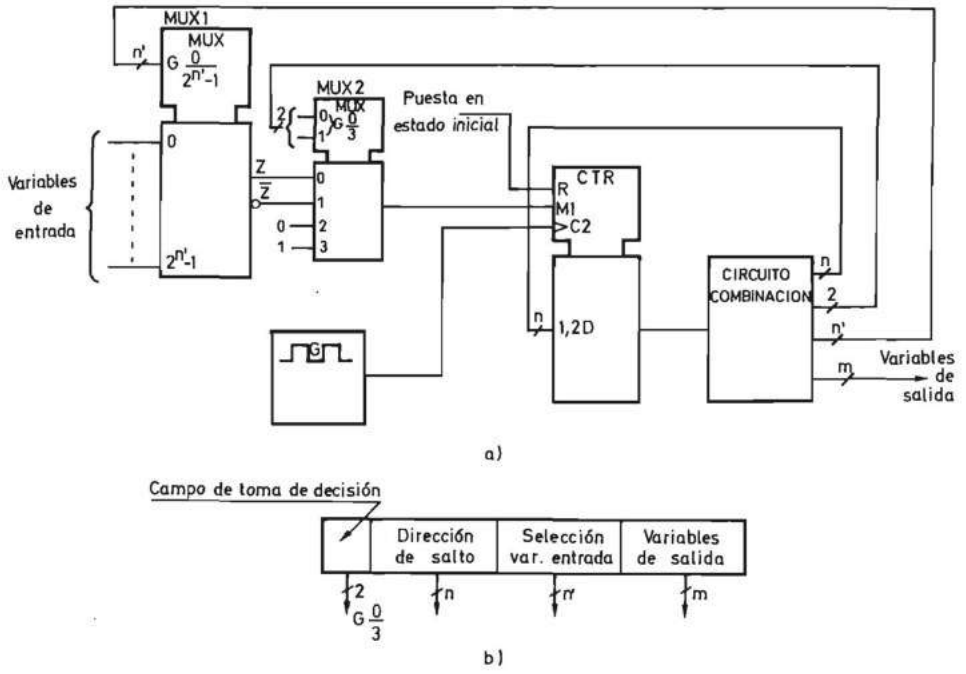


FIGURA 6.271.—**Sistema secuencial síncrono realizado con un multiplexor de las variables de entrada (MUX1) y un multiplexor de toma de decisión (MUX2): *a*) esquema del circuito; *b*) vector de salida del circuito combinacional.**

representa el vector de salida del circuito combinacional de este sistema secuencial síncrono, que presenta la ventaja de hacer posible la realización modular del multiplexor MUX1.

En efecto, una de las formas de realizar los multiplexores, estudiada en el apartado 3.7.3.2, es mediante un conjunto de puertas seguidoras de tres estados que tienen sus entradas de desinhibición controladas por un decodificador y sus salidas conectadas a una barra (bus). En la figura 6.272 se representa el multiplexor de la figura 6.271 realizado de esta forma.

Para lograr la modularidad, las $2^{n'}$ variables de entrada se dividen en 2^m grupos o módulos de $2^{n'-m}$ variables cada uno. Pero además es necesario que los módulos se puedan conectar entre sí y que no se seleccione más de un módulo en cada instante. Para ello las m variables que definen el grupo se comparan con la combinación generada por un conjunto de interruptores y la salida de este comparador se conecta a la entrada de desinhibición (*EN*) del decodificador que genera las variables de desinhibición de las puertas conectadas a la barra de entrada (figura 6.273).

En la figura 6.274 se representa el diagrama de bloques del módulo de la figura 6.273 y en la figura 6.275 la unidad modular de entrada formada por los 2^m módulos programados interiormente para corresponder a los grupos cero al $2^m - 1$.

Pero además de modularidad de las variables de entrada, para que un sistema secuencial síncrono sea verdaderamente modular ha de poseer modularidad de las va-

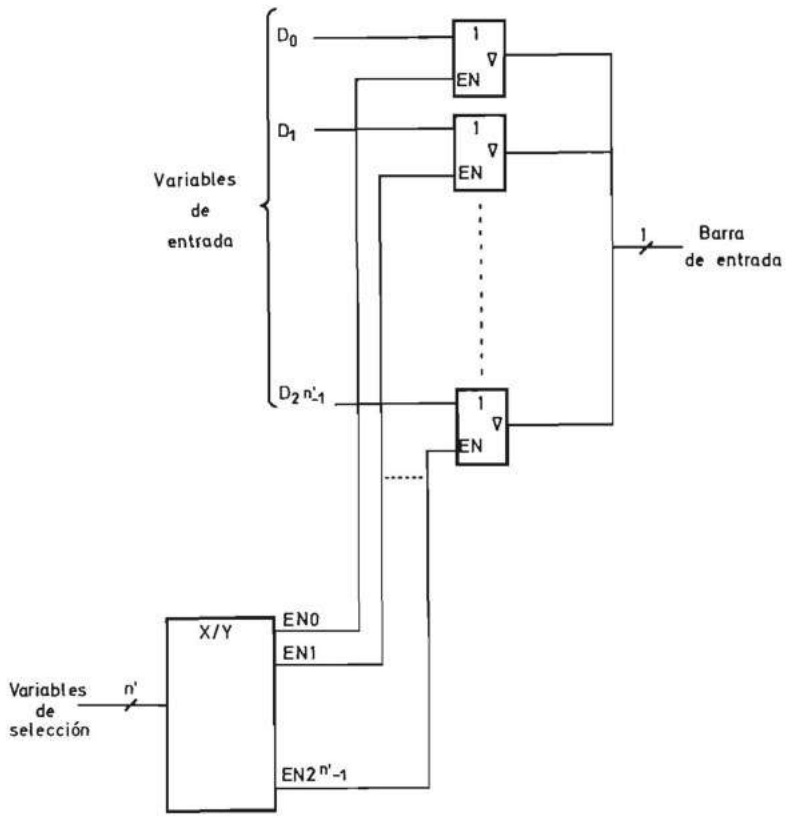


FIGURA 6.272.—**Esquema de un multiplexor de las variables de entrada realizado con puertas de tres estados.**

riables de salida. Esta modularidad resulta especialmente útil en la aplicación de los sistemas secuenciales síncronos a la realización de controladores lógicos, que se analiza en el apartado 6.3.5.6.3.

Para realizar un sistema secuencial síncrono con modularidad de las variables de salida, se utiliza una sola variable binaria de salida del circuito combinacional como variable de salida del sistema secuencial síncrono y se memoriza su estado en un conjunto de biestables que constituyen la unidad de salida representada en la figura 6.276, que puede ser realizada de forma modular al igual que la de entrada. Al aplicar los flancos activos, en este caso los de subida, del generador de impulsos a la entrada *C*2 de todos los biestables, la información presente en la entrada *D* (1,2*D*) de todos ellos se introduce solamente en aquel cuya entrada *G*1 se encuentra en nivel uno.

En la figura 6.277 se representa el símbolo correspondiente al circuito de la figura 6.276 que se utiliza en la figura 6.278 para formar parte de un sistema secuencial síncrono que posee modularidad de entradas y de salidas. La unidad de salida

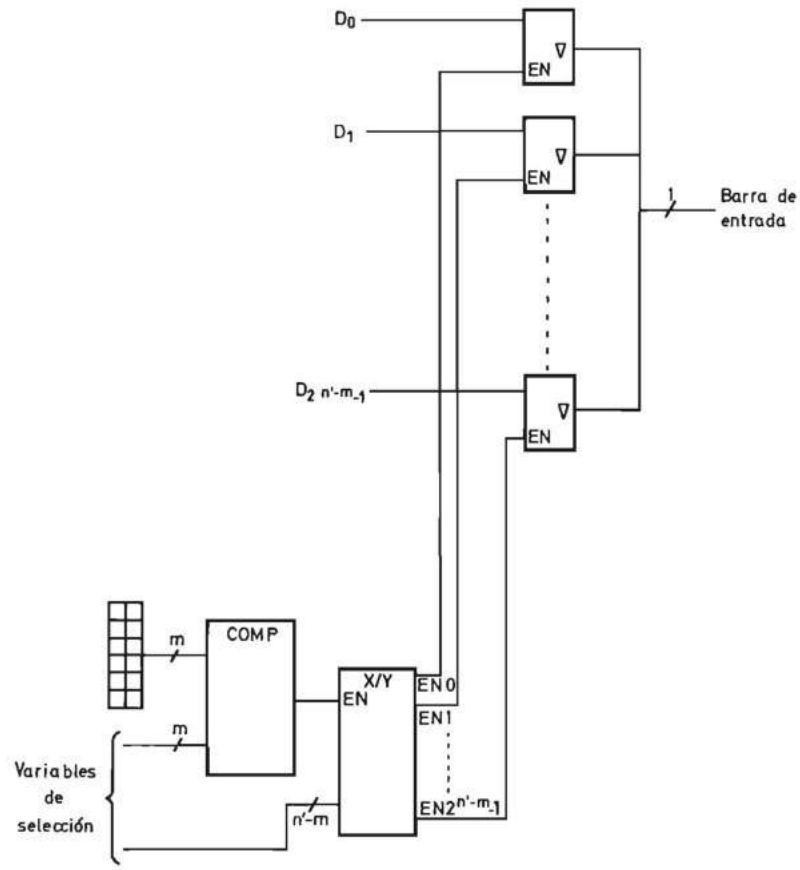


FIGURA 6.273.—**Módulo de $2^{n'-m}$ variables de entrada realizado con un conjunto de interruptores (microswitches) y un comparador.**

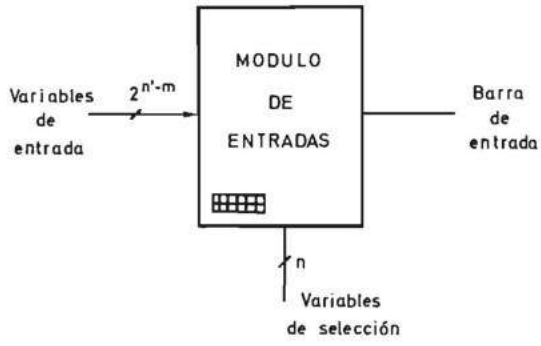


FIGURA 6.274.—**Diagrama de bloques del módulo de la figura 6.273.**

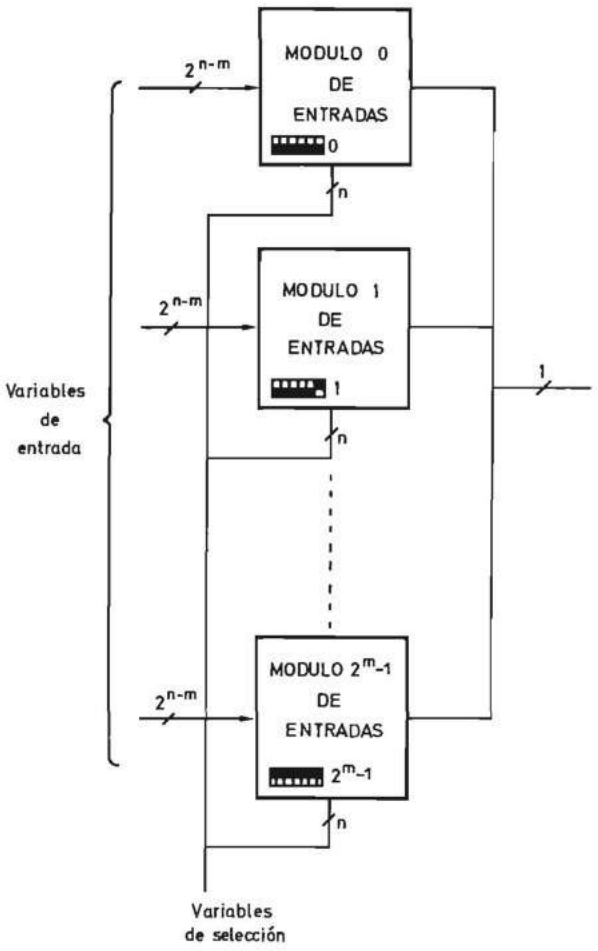


FIGURA 6.275.—**Unidad de entrada de un sistema secuencial síncrono modular.**

posee una entrada de inhibición $G1$ para que la información sólo entre en el biestable seleccionado en aquellas microinstrucciones en que así se desee. En la figura 6.278 se representa el vector de salida del circuito combinacional.

6.3.5.5.5 Sistemas secuenciales síncronos semimodulares. Los sistemas secuenciales síncronos modulares son interesantes cuando se prevén cambios a lo largo de la vida útil de los mismos, lo cual compensa el mayor coste de su sistema físico que es más complejo que el de los no modulares.

Pero el interés de introducir los sistemas secuenciales síncronos programables en un único circuito integrado normalizado disponible comercialmente hizo surgir el interés por los sistemas secuenciales síncronos semimodulares, cuyo esquema básico, que se obtiene a partir del de la figura 6.250, se representa en la figura

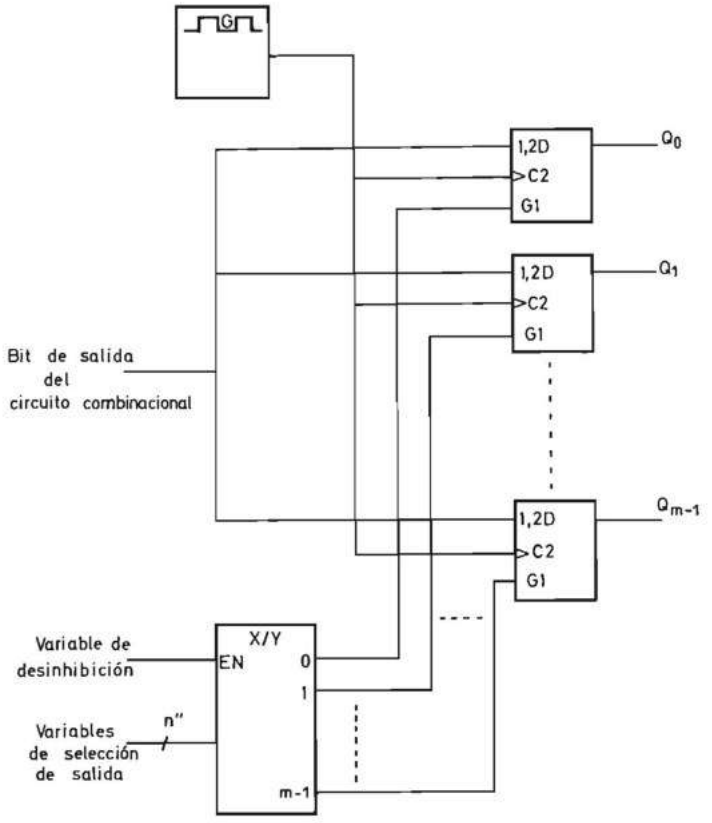


FIGURA 6.276.—**Unidad de salida de un sistema secuencial síncrono modular.**

6.279. El lector puede comprobar que la diferencia entre ambas consiste en la existencia en la figura 6.279 de puertas seguidoras de tres estados cuyas entradas se conectan a las variables de salida del circuito combinacional y sus salidas a la entrada de este último.

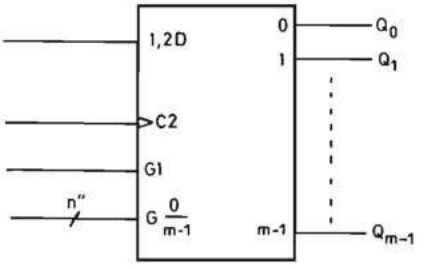


FIGURA 6.277.—**Símbolo lógico correspondiente a la unidad de salida de la figura 6.276.**

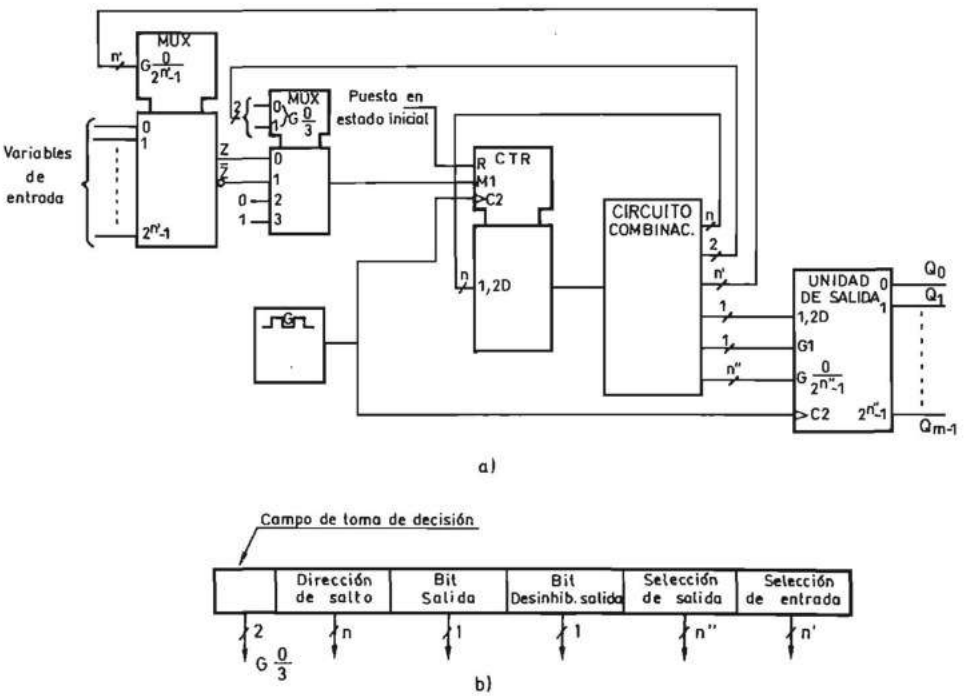


FIGURA 6.278.—**Esquema de un sistema secuencial síncrono con modularidad de entrada y de salida.**

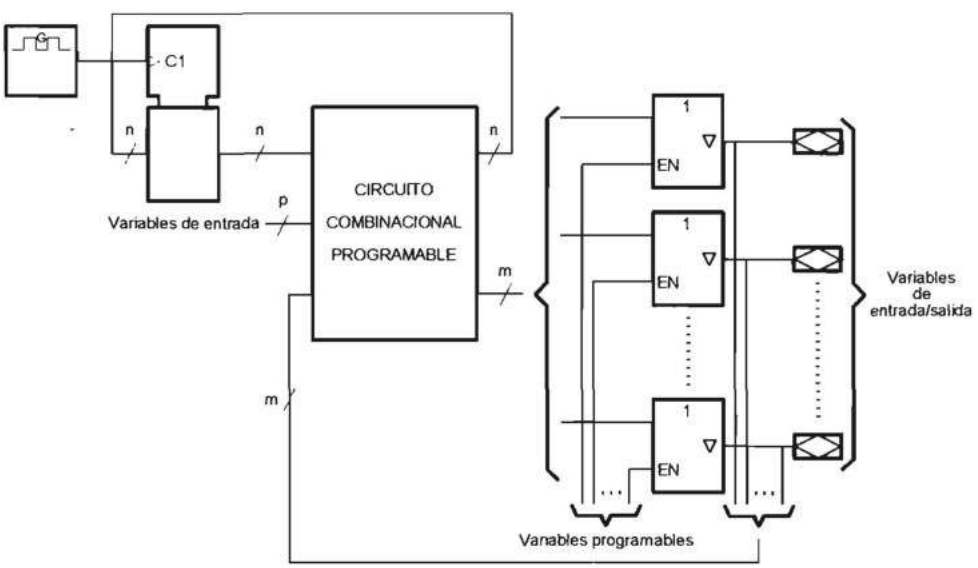


FIGURA 6.279

Las puertas seguidoras de tres estados poseen entradas de desinhibición (ENABLE) independientes que se pueden poner a nivel cero o a nivel uno. En el caso de que una variable *EN* esté en nivel uno, la salida de la puerta seguidora correspondiente no está en tercer estado y la variable binaria asignada a ella es una variable de salida que se lleva al exterior a través del terminal al que está conectada la puerta.

Por el contrario, si una variable *EN* está en nivel cero, la salida de la puerta seguidora se pone en tercer estado y al terminal correspondiente se puede conectar una variable de entrada.

El sistema digital de la figura 6.279 constituye un bloque funcional realizable en circuito integrado. En particular tiene gran interés su realización como circuito normalizado comercial (en inglés «Standard off the shelf integrated circuit»), que ha recibido el nombre genérico de «Dispositivo Lógico Programable» (DLP) [en inglés «Programmable Logic Device (PLD)]. Este tipo del circuitos ha sido posible solamente gracias al progreso de las técnicas de integración y su desarrollo ha sido espectacular a partir de 1990. Por ello su estudio se sale fuera de los límites de este libro y al lector interesado se le remite a la bibliografía [MAND 92].

6.3.5.6 Aplicaciones de los sistemas secuenciales síncronos de control

6.3.5.6.1 Introducción. En los apartados anteriores se estudiaron los diferentes sistemas secuenciales síncronos como, por ejemplo, los contadores, registros de desplazamiento, conjuntos de registros, etc., que permiten realizar sistemas que generan variables binarias cuyo valor en un instante determinado no depende solamente del valor que tienen las variables de entrada en ese momento, sino del que tuvieron en el pasado. Se obtuvieron así los sistemas secuenciales síncronos que evolucionan en función de la secuencia de estados de las variables de entrada.

En sucesivos apartados se analizan los dos grandes campos de aplicación de los sistemas secuenciales síncronos:

a) Los procesadores digitales secuenciales.
b) Los controladores lógicos.

6.3.5.6.2 Procesadores digitales secuenciales. Mediante sistemas combinacionales que constituyen operadores aritméticos o lógicos se puede realizar cualquier proceso de información. Un sistema de este tipo recibe el nombre de procesador combinacional y su forma de operación se define mediante una tabla de verdad. En los capítulos 3 y 4 se estudian varios ejemplos de procesadores combinacionales: circuitos comparadores de dos combinaciones binarias (apartado 3.7.4), sumadores/restadores de números codificados en binario natural (apartado 4.2.3.2) o en BCD natural (apartado 4.3), unidades aritméticas y lógicas (apartado 4.2.4), etc.

Combinando bloques funcionales combinacionales se puede realizar cualquier procesador. Para mostrarlo se elige un ejemplo que permite exponer las características de los procesadores digitales secuenciales.

Ejemplo 6.9:

Realizar un procesador combinacional que compare tres números binarios positivos A, B y C codificados en binario natural de ocho bits y dé como resultado el mayor de los tres. Utilícense para ello bloques funcionales comparadores y multiplexores.

La solución se representa en la figura 6.280 y consta de dos comparadores de ocho bits y dos multiplexores compuestos cada uno, a su vez, de ocho multiplexores de dos canales de entrada. El primer comparador COMP 1 realiza la comparación de A y B. Al mismo tiempo ambas combinaciones se conectan a las entradas del multiplexor MUX 1 compuesto por ocho multiplexores de dos canales de entrada y cuya variable de selección $G1$ se conecta a la salida $P > Q$ del comparador. A la salida del multiplexor MUX 1 aparece de esta forma la combinación mayor de A y B, que se conecta a una de las entradas de otro comparador COMP 2 idéntico al COMP 1. A otra de las entradas del COMP 2 se presenta el tercer operando C y, finalmente, se conecta a otro multiplexor MUX 2 idéntico al MUX 1, a cuya salida aparece el resultado que es el máximo de los tres operandos A, B y C. La realización práctica se puede llevar a cabo con bloques funcionales integrados en escala de integración media.

Pero los procesadores combinacionales presentan varias limitaciones que pueden observarse en el ejemplo anterior:

a) Hacen necesario que se disponga simultáneamente de todos los operandos, lo cual en numerosas situaciones prácticas no es factible.
b) La realización de operaciones complejas implica la utilización de circuitos com-

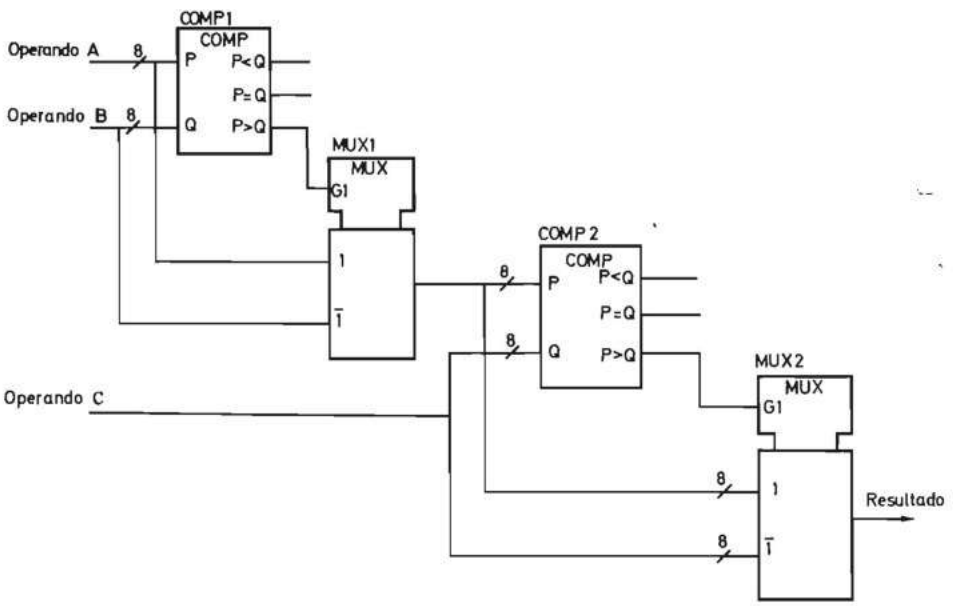


FIGURA 6.280.—**Ejemplo de procesador digital combinacional.**

binacionales complejos. Esta limitación se acentúa si, además, el procesador debe realizar varias operaciones complejas diferentes.

Ello trae consigo el interés de la utilización de los procesadores secuenciales que ejecutan una operación compleja mediante una secuencia de operaciones elementales realizadas por un circuito combinacional sencillo que, para la ejecución de operaciones aritméticas, es un simple sumador/restador. Su utilización en secuencia permite la ejecución de multiplicaciones y divisiones, y mediante desarrollos en serie cualquier operación matemática como, por ejemplo, una función trigonométrica. En la práctica, el interés de operar con información alfanumérica o simples variables binarias, hace que el circuito combinacional utilizado sea una unidad aritmética y lógica como la estudiada en el apartado 4.2.4.

La ejecución de una operación compleja mediante una secuencia de operaciones elementales realizadas por un circuito combinacional sencillo implica:

a) La necesidad de que el procesador posea capacidad de memorización de resultados parciales.
b) La selección de los operandos, las operaciones elementales a realizar y los resultados parciales.

La primera exigencia se consigue mediante un conjunto de registros.

Cuando la cantidad de información (datos y resultados parciales) a memorizar es pequeña (hasta algunos cientos de bits), se utilizan simples registros de almacenamiento en serie o en paralelo. Cuando la cantidad de información es elevada, los registros se organizan constituyendo unidades de memoria cuyo estudio se realiza en el capítulo 7.

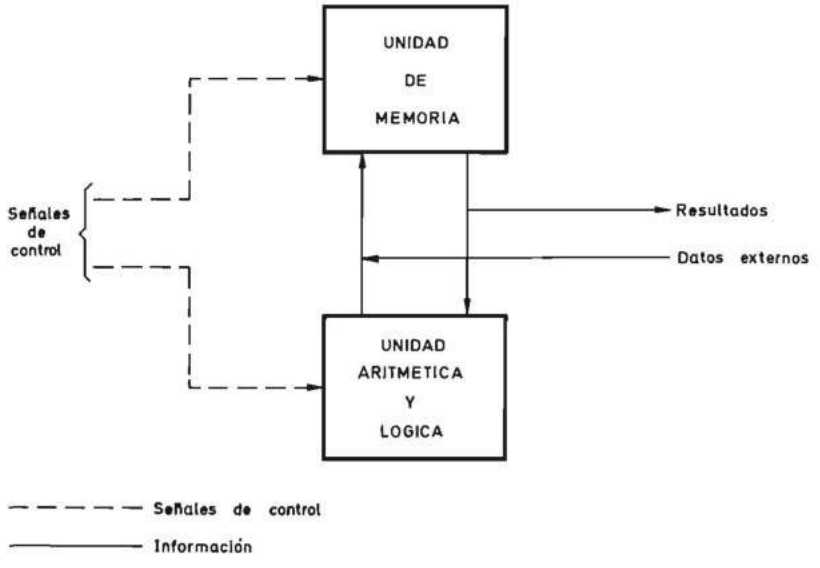


FIGURA 6.281.—**Partes que constituyen una unidad operativa.**

Aunque físicamente la unidad aritmética y lógica y la unidad de memoria están a veces entremezcladas, sus elementos se pueden distinguir perfectamente.

Los datos externos se memorizan en los registros adecuados de la unidad de memoria para presentarlos en secuencia en las entradas de la unidad aritmética que los procesa y devuelve a la unidad de memoria los resultados parciales y finales obtenidos. Estos últimos se presentan de alguna manera al sistema externo que entregó los datos (fig. 6.281).

El conjunto formado por una unidad de memoria y la unidad aritmética y lógica puede recibir el nombre de unidad operativa y ser representado mediante un único bloque tal como se indica en el figura 6.282.

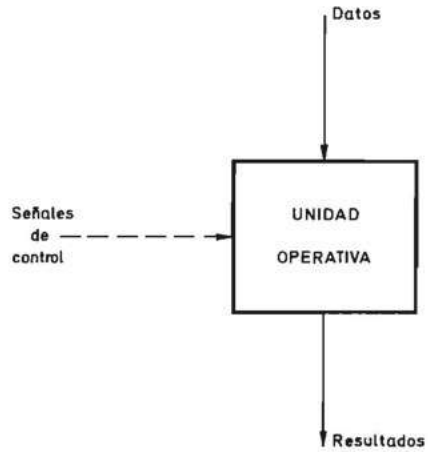


FIGURA 6.282.—**Diagrama de bloques de la unidad operativa.**

La segunda exigencia que debe cumplimentar un procesador digital secuencial se consigue mediante una unidad de control que genere en secuencia las señales que seleccionan en cada instante los operandos y la operación a realizar con ellos.

Tal como se estudia en los capítulos 7 y 4 respectivamente, las unidades de memoria poseen entradas de control y las unidades aritméticas y lógicas entradas de selección de operación. Ambos tipos de entradas son gobernadas mediante un sistema digital que genera una secuencia de señales de control (secuencias de ceros y unos de duración y relación adecuadas en el tiempo). Este sistema se denomina unidad de control y constituye un sistema secuencial como los estudiados a lo largo de este capítulo.

Se obtiene de esta forma el diagrama de bloques de un procesador digital secuencial representado en la figura 6.283.

La forma de realizar la unidad de control permite clasificar a su vez los procesadores en dos clases:

a) Procesadores digitales síncronos en los que la unidad de control es un sistema secuencial síncrono. En estos procesadores la secuencia de microoperaciones

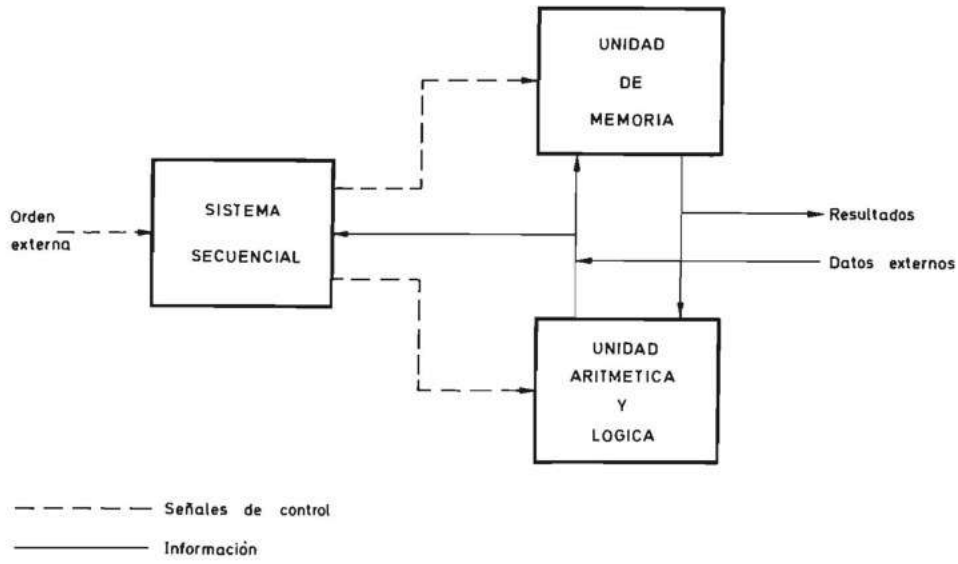


FIGURA 6.283.—**Diagrama de bloques de un procesador digital secuencial.**

que forman una operación y el concatenamiento de éstas se efectúa por medio de los impulsos de un reloj o generador de impulsos interno. Esto permite pasar de una microoperación a la siguiente sin necesidad de detectar que la primera ha acabado. Simplemente hay que prever que la separación entre flancos activos del generador de impulsos sea suficiente para que se realice la microoperación.

b) Procesadores digitales asíncronos en los que la unidad de control es un sistema secuencial asíncrono. En estos procesadores no existe un generador de impulsos o reloj y el concatenamiento de las operaciones se realiza detectando que ha terminado una para dar orden de iniciar la siguiente. La principal ventaja de los sistemas asíncronos es que la velocidad de operación es la máxima que permite la tecnología utilizada, pero la complejidad que se añade al sistema al tener que realizar los detectores de fin de operación hace que los procesadores digitales secuenciales sean en su casi totalidad síncronos. En éstos, la velocidad de operación es menor porque la separación entre impulsos ha de ser tal que dé tiempo a realizarse la microoperación más lenta. Esta dificultad se ha ido superando mediante la utilización de tecnologías cada vez más rápidas.

Por todo lo expuesto los procesadores digitales son en la práctica síncronos y su diagrama de bloques es representado en la figura 6.284.

Fusionando la unidad de memoria y la unidad aritmética y lógica en un único bloque denominado unidad operativa, tal como vimos anteriormente, se obtiene finalmente el diagrama de bloques de un procesador digital secuencial representado en la figura 6.285 en la que, para abreviar, se le ha quitado a la unidad de con-

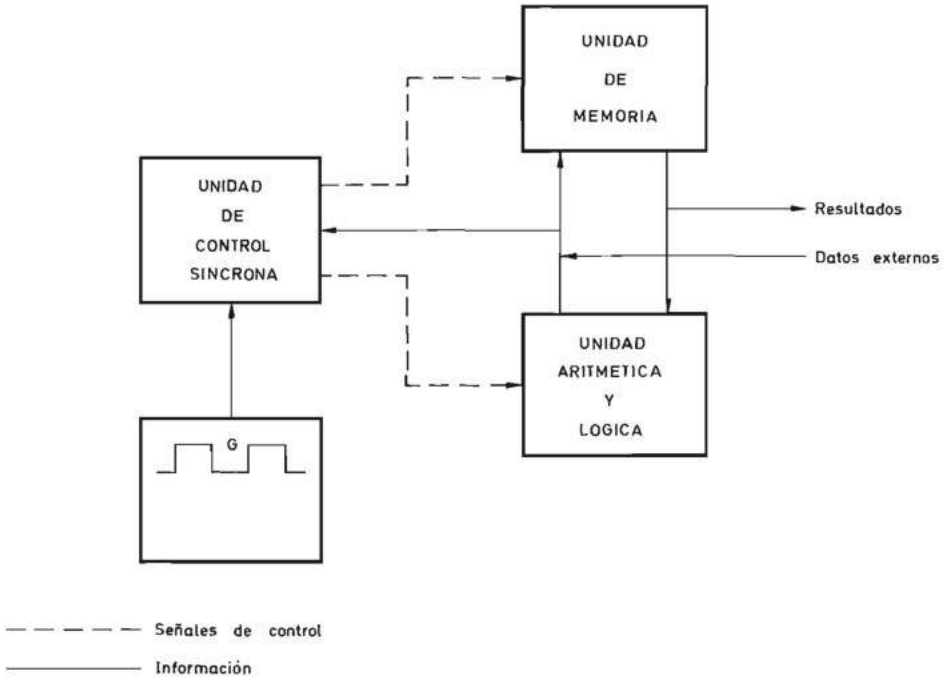


FIGURA 6.284.—**Diagrama de bloques de un procesador digital secuencial síncrono.**

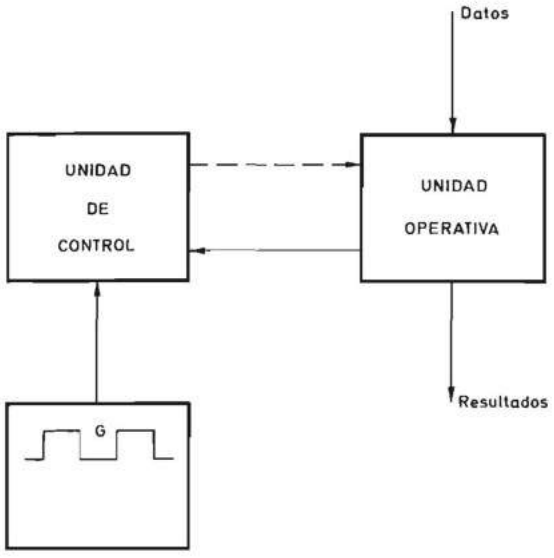


FIGURA 6.285.—**Diagrama de bloques de un procesador digital secuencial síncrono.**

trol el apelativo de síncrona porque el generador de impulsos ya indica dicha circunstancia.

Como ejemplo de todo lo expuesto se puede convertir en secuencial el procesador de la figura 6.280.

Ejemplo 6.10:

Realizar la unidad operativa de un procesador secuencial que compare tres números binarios A, B y C codificados en binario natural de ocho bits, y dé como resultado el mayor de los tres. Utilícese para ello un único bloque funcional comparador.

En la figura 6.286 se representa el esquema del circuito correspondiente en el que se utiliza un único comparador (COMP1) y un multiplexor (MUX1) en lugar de los dos de la figura 6.280.

Para ello ha sido necesario utilizar además los siguientes elementos:

— Un registro *R*1 conectado a la salida de MUX1 en el que se almacena en el instante adecuado el mayor de los números presentes en la entrada del comparador COMP1.
— Un doble multiplexor MUX2 de dos canales para seleccionar los números que se presentan a la entrada del comparador COMP1.

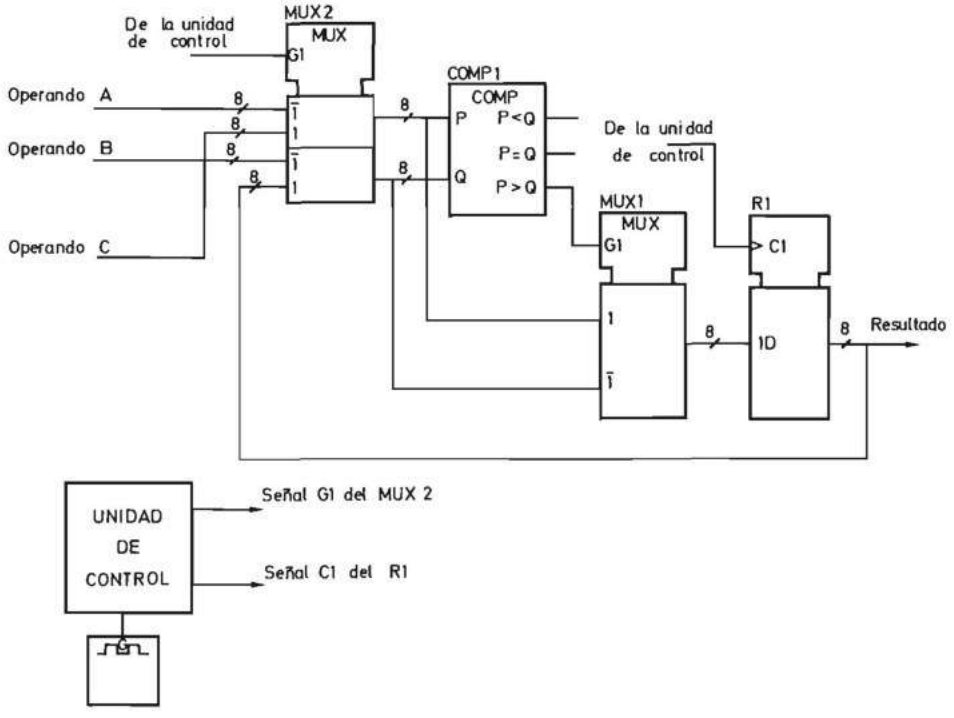


FIGURA 6.286.—**Ejemplo de procesador digital secuencial.**

En primer lugar es necesario poner la entrada $G1$ del multiplexor MUX2 en nivel cero para que a las entradas del COMP1 se presenten los operandos A y B. Al mismo tiempo es necesario aplicar un impulso en la entrada de control $C1$ de $R1$ para que en él se memorice el mayor de ambos. A continuación es necesario pasar $G1$ a nivel uno para que a las entradas del COMP1 se presenten el operando C y el contenido de $R1$ y volver a aplicar un impulso a la entrada $C1$ de éste para que en él se almacene el mayor de los tres números.

Las señales de control $G1$ y $C1$ las genera una unidad de control que constituye un sistema secuencial síncrono.

De todo lo expuesto se deduce que la realización de procesadores digitales secuenciales constituye una de las aplicaciones más importantes de los sistemas secuenciales síncronos de control y por eso a su estudio se dedica el tomo II.

6.3.5.6.3 Controladores lógicos síncronos. Los controladores lógicos son sistemas digitales que actúan sobre una o más variables binarias de salida, en función del estado o del cambio de estado de unas variables binarias de entrada.

Desde este punto de vista constituyen controladores lógicos los sistemas combinacionales estudiados en el capítulo 3 y los sistemas secuenciales asíncronos estudiados en el apartado 6.2. Pero tanto unos como otros presentan serias limitaciones en su aplicación práctica. Los sistemas combinacionales no pueden memorizar las variables de entrada para tomar decisiones posteriormente. Por su parte, los sistemas secuenciales asíncronos presentan problemas de fenómenos aleatorios cuando cambian de estado varias variables de entrada simultáneamente y no admiten aumento del número de variables de entrada o salida sin modificar el sistema físico.

Por todo ello surge el interés de los controladores lógicos síncronos que presentan las siguientes características:

a) Ausencia de fenómenos aleatorios cuando cambian varias variables de entrada o de estado interno simultáneamente.
b) Posibilidad de modificar su forma de operación sin necesidad de cambiar su cableado.
c) Posibilidad de modularidad o semimodularidad de variables de entrada y salida.

El esquema básico de un controlador lógico síncrono se representa en la figura 6.287 que coincide con el sistema secuencial síncrono de las figuras 6.6 o 6.155 al que se ha añadido un registro $R2$ de sincronización de las variables de entrada. Este registro tiene como finalidad hacer que las variables de entrada conectadas al sistema combinacional, estén estables en el momento en que se apliquen a la entrada $C1$ del registro $R1$ los flancos activos del generador de impulsos, para evitar de esta forma la aparición de fenómenos aleatorios.

El registro $R2$ de sincronización puede ser sustituido por dos registros síncronos activados por niveles inversos conectados en serie (en un montaje principal-subordinado). Se obtiene de esta forma el controlador lógico síncrono de la figura 6.288 cuyo comportamiento es idéntico al de la figura 6.287.

Por otra parte, es posible realizar cualquier controlador lógico síncrono de tal

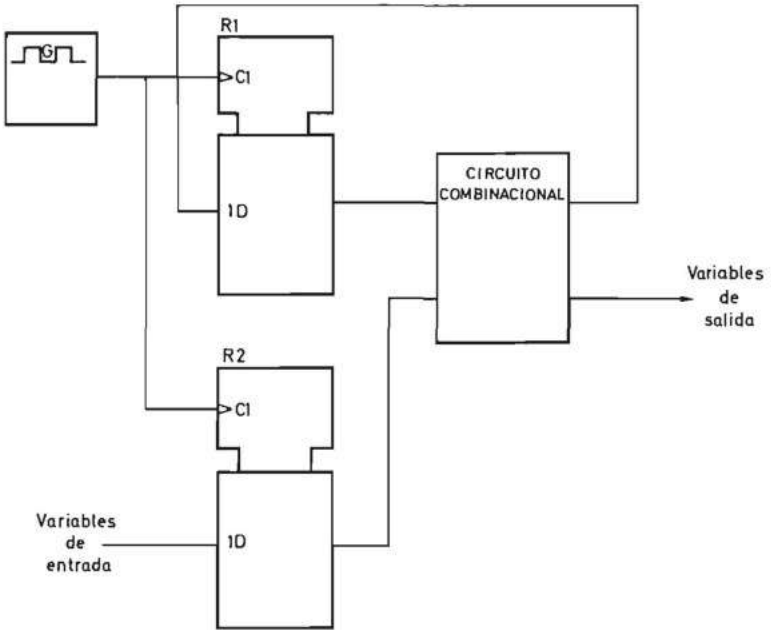


FIGURA 6.287.—**Esquema básico de un controlador lógico síncrono con un registro de sincronización activado por flancos.**

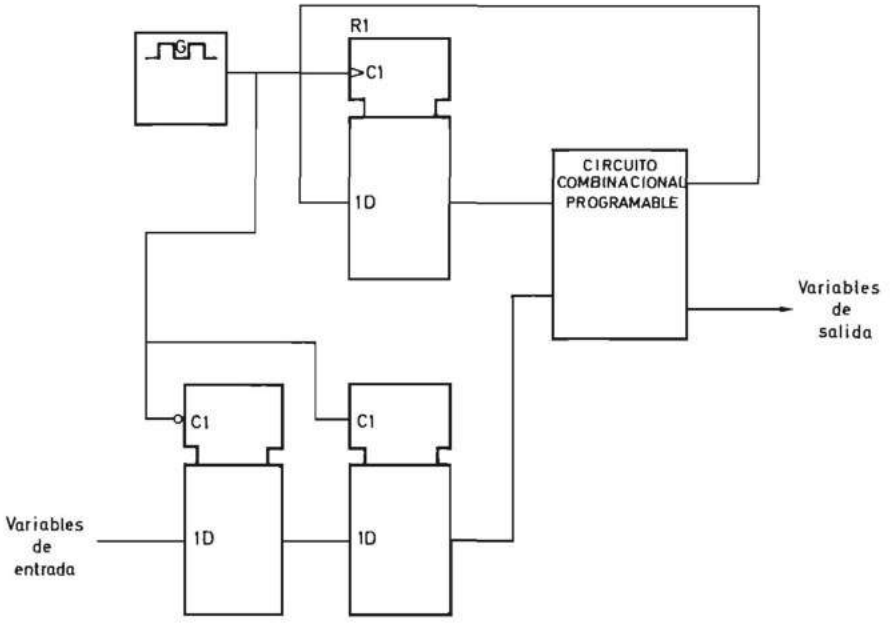


FIGURA 6.288.—**Esquema básico de un controlador lógico síncrono con dos registros de sincronización activados por niveles.**

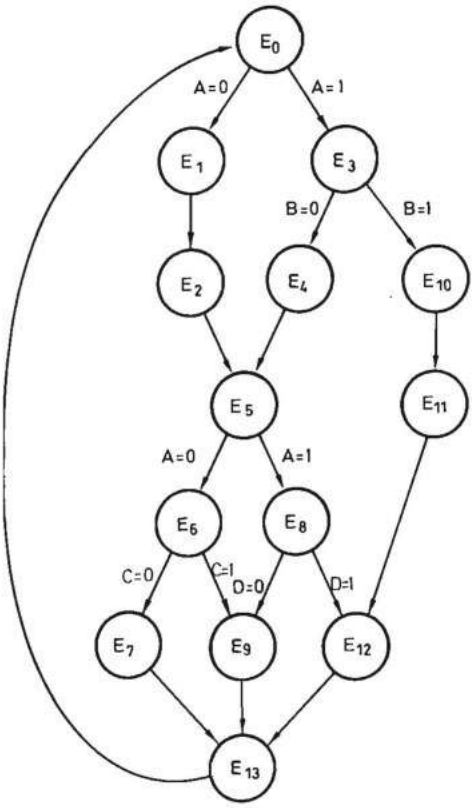


FIGURA 6.289.—**Ejemplo de diagrama de flujo de un controlador lógico síncrono.**

manera que en su diagrama de flujo las tomas de decisión se realicen entre un estado consecutivo al actual en binario natural y otro no consecutivo. En la figura 6.289 se representa un ejemplo de diagrama de flujo. De esta forma, se puede sustituir el registro $R1$ de las figuras 6.287 y 6.288 por un contador, obteniéndose el esquema de un controlador lógico representado en la figura 6.290.

Los controladores lógicos de las figuras 6.287, 6.288 y 6.290 pueden observar varias variables de entrada simultáneamente y realizar cualquier función lógica en un único estado interno. Pero también es posible conseguir el mismo efecto mediante la observación de una sola variable de entrada en cada instante combinada con las tomas de decisión adecuadas. Como ejemplo, en la figura 6.291*a* y *b* se representan los diagramas de flujo correspondientes a la realización de las funciones $f_1 = abc$ y $f_2 = a+b+c$ respectivamente.

De lo expuesto se deduce que es posible realizar un controlador lógico síncrono de acuerdo con el esquema de la figura 6.292*a* en el que en lugar de aplicar las variables de entrada al circuito combinacional a través del registro R (figura 6.290), se conectan a las entradas de un multiplexor MUX1 cuyas variables de selección son controladas por las correspondientes salidas del circuito combinacional (figura

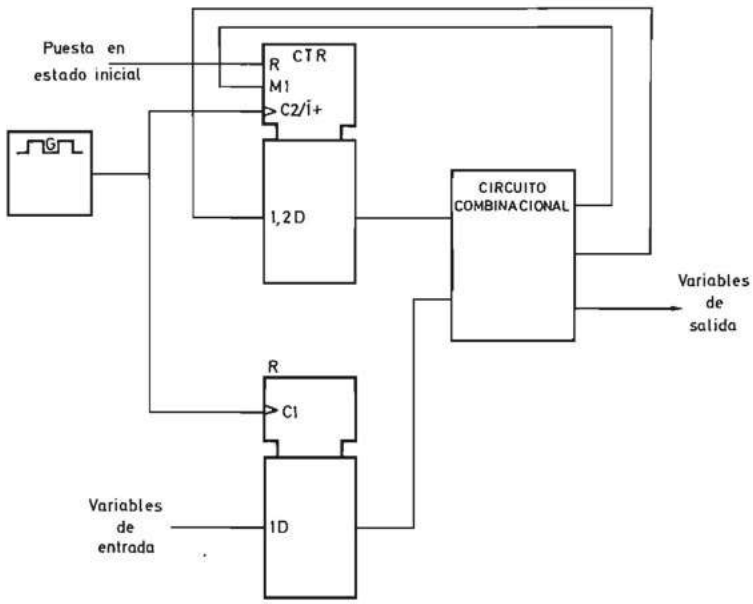


FIGURA 6.290.—**Esquema de un controlador lógico síncrono realizado con un contador.**

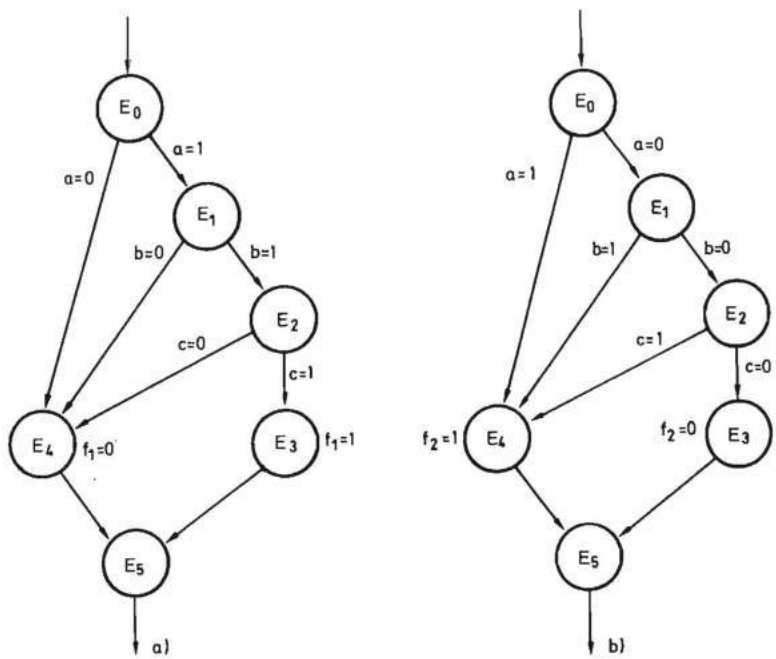


FIGURA 6.291.—**Realización de las funciones lógicas $f_1 = abc$ y $f_2 = a+b+c$ mediante un controlador lógico que posee un multiplexor de las variables de entrada.**

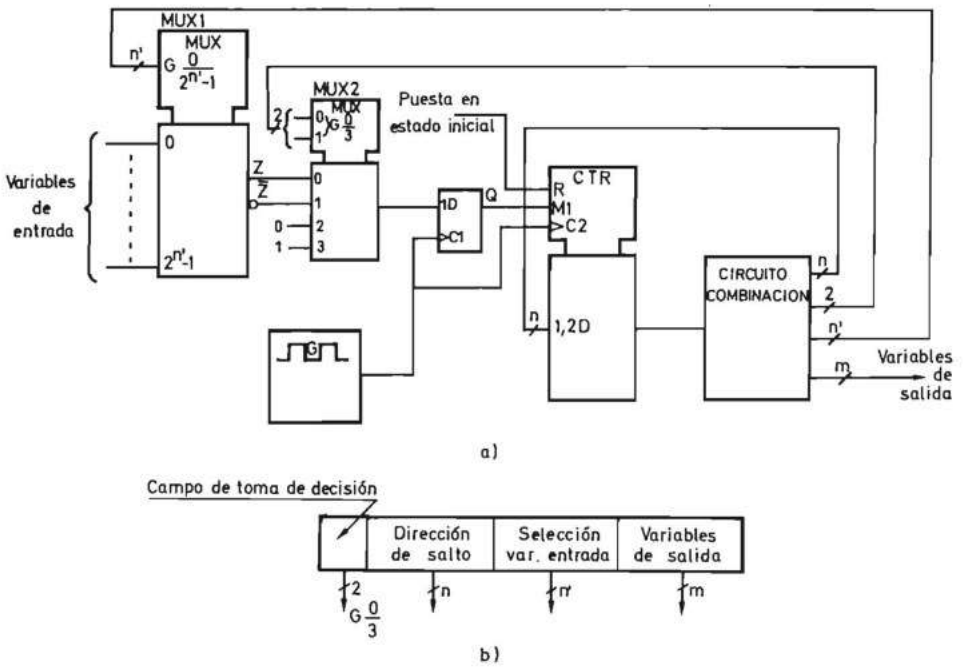


FIGURA 6.292.—**Controlador lógico síncrono realizado con un multiplexor de las variables de entrada (MUX1) y un multiplexor de toma de decisión (MUX2): *a*) esquema del circuito; *b*) vector de salida del circuito combinacional.**

6.292). El lector puede comprobar que el esquema de la figura 6.292 se diferencia del de la figura 6.271 solamente en que posee un biestable *D* activado por flancos que sincroniza las variables de entrada con los flancos activos del generador de impulsos.

6.4 REPRESENTACION GRAFICA DE LAS SEÑALES DE UN SISTEMA SECUENCIAL SINCRONO

Tal como se estudió en los apartados anteriores, las variables de salida de un sistema secuencial no dependen solamente del estado de las variables de entrada en un instante determinado, sino que en ellas influye la secuencia de evolución de las variables de entrada. Además, en los sistemas secuenciales síncronos, los cambios de estado interno se producen en sincronismo con los flancos activos (cambios de subida o bajada) de una onda cuadrada procedente de un generador de impulsos.

Por todo lo expuesto, una de la mejores maneras de expresar la forma de comportamiento de un sistema secuencial síncrono es mediante un diagrama de evolución en el tiempo de las señales de entrada y salida adecuadamente relacionadas entre sí.

Cuando el número de señales de un sistema secuencial es reducido, no existe mayor dificultad para realizar el diagrama de evolución. Pero cuando dicho número se eleva y como consecuencia lo hacen también las relaciones entre las distintas señales, es conveniente establecer unas normas de representación que faciliten la interpretación de dichos diagramas de evolución en el tiempo, que reciben también el nombre de cronogramas.

No existe una normalización universalmente aceptada al respecto aunque se han realizado diversas propuestas sobre este tema [OSBO 81] [RONY 80].

A continuación estudiaremos las principales relaciones entre las señales de un sistema secuencial síncrono e indicaremos una forma de representarlas gráficamente, la cual será utilizada en el resto del libro.

a) Relaciones entre dos señales.

Tal como se estudió en apartados anteriores, una señal de un sistema secuencial puede actuar sobre el mismo por flancos o por nivel. De ahí que se puedan dar diversas situaciones.

En la figura 6.293 se representa la forma de especificar que el cambio de una señal provoca el cambio de otra. Para ello se coloca un círculo en el medio del flanco de ambas y una flecha dirigida desde la primera a la segunda.

En la figura 6.294 se representa la forma de especificar que la puesta en un determinado nivel de una señal provoca un nivel en otra. Para ello se coloca un círculo en el nivel de la primera y otro círculo en el nivel de la segunda enlazados por un segmentos dirigido de la primera a la segunda.

En la figura 6.295 se indica la forma de especificar que la puesta en un determinado nivel de una señal hace que otra cambie de estado, y en la figura 6.296 la situación inversa.

b) Relación entre varias señales.

En ocasiones el cambio o la colocación en un determinado nivel de una señal depende de más de una. Dicha situación se representa de forma similar al caso anterior mediante los segmentos dirigidos adecuados. En la figura 6.297 se representa como ejemplo la forma de especificar que la señal x_3 cambia de cero a uno cuando x_1 se encuentra en nivel cero y x_2 cambia de cero a uno.

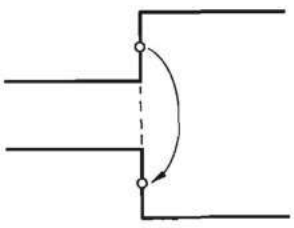


FIGURA 6.293—**Especificación de que el cambio de una señal provoca el cambio de otra.**

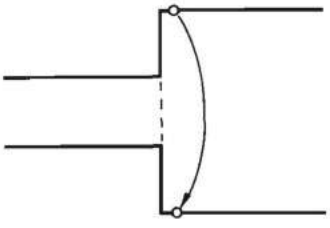


FIGURA 6.294—**Especificación de que la puesta en un determinado nivel de una señal provoca un nivel en otra.**

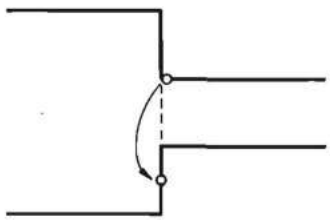


FIGURA 6.295.—**Especificación de que la puesta en un determinado nivel de una señal provoca un cambio en otra.**

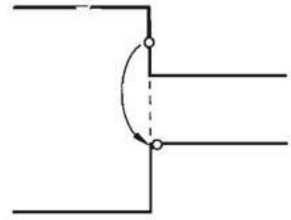


FIGURA 6.296.—**Especificación de que el cambio de estado de una señal provoca un nivel en otra.**

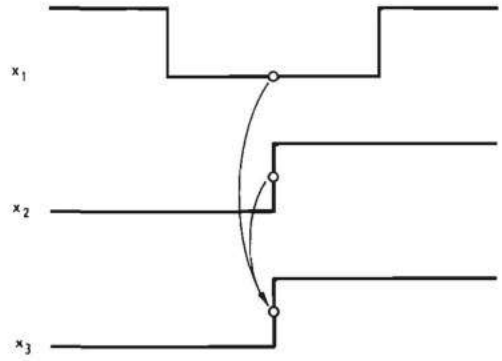


FIGURA 6.297.—**Especificación de que la simultaneidad de un nivel de una señal (x_1) y un cambio en otra (x_2) provoca un cambio en una tercera (x_3).**

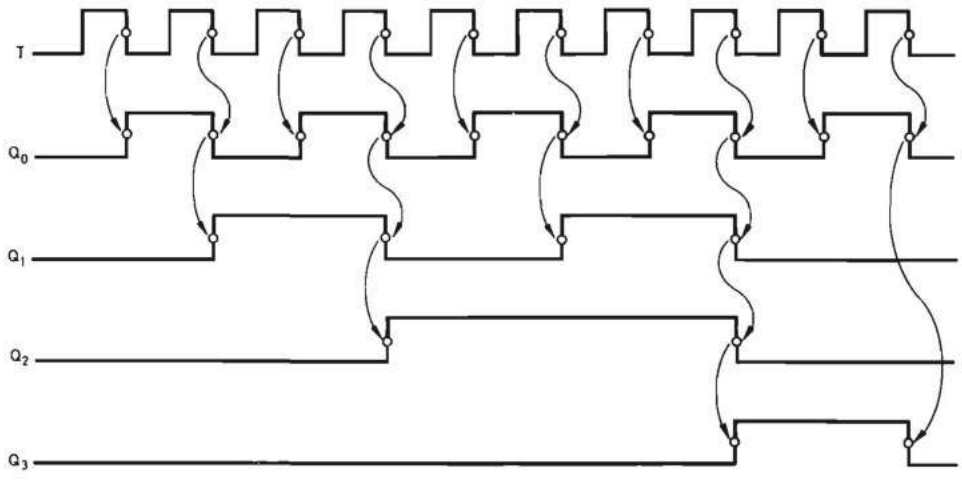


FIGURA 6.298.—**Especificación del diagrama de evolución temporal de las señales de un contador BCD natural.**

Como ejemplo práctico de todo lo que acabamos de exponer, en la figura 6.298 se representa el diagrama de evolución de un contador en BCD natural asíncrono que ya se indicó en la figura 6.198 sin utilizar los convenios establecidos.

c) Relación entre una o más señales independientes y un conjunto de señales.

En procesadores digitales que operan en paralelo con un cierto número de bits, es usual que se tengan que representar acciones conjuntas sobre el estado de todos ellos (tomo II), sin que tenga importancia el estado concreto individual que depende del valor de un dato o de un resultado obtenido mediante una determinada operación aritmética o lógica.

La forma de especificar gráficamente el cambio de un conjunto de bits se representa en la figura 6.299.

En lugar de representar cada señal por separado se utiliza una representación conjunta. La indicación de que algunos bits cambian de cero a uno y otros de uno a cero, se realiza mediante las señales de ambos niveles que se cortan en el instante en que se produce el cambio.

El cambio de estado de los bits de una combinación binaria (por ejemplo, el resultado de una operación) se produce, en general, con motivo del cambio de estado de una señal binaria. En la figura 6.300 se representa gráficamente la situación de que el cambio de estado de la variable Q produce el cambio de estado de la información binaria formada por las variables D_0 a D_7.

En ocasiones, el estado de las variables que constituyen la información es indiferente antes del cambio. En este caso las líneas de ambos niveles se indican como discontinuas (fig. 6.301).

En la figura 6.302 se representa un ejemplo práctico. En la figura 6.302*a* se

FIGURA 6.299.—**Especificación del cambio de un conjunto de bits.**

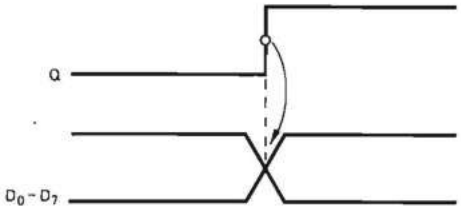


FIGURA 6.300.—**Especificación de que el cambio de un conjunto de bits se produce como consecuencia del cambio de estado de una señal binaria.**

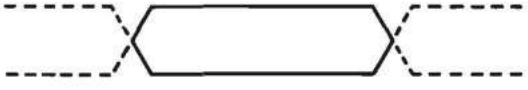


FIGURA 6.301.—**Especificación de un conjunto de bits que permanece en un cierto estado durante un determinado intervalo y es indiferente fuera de él.**

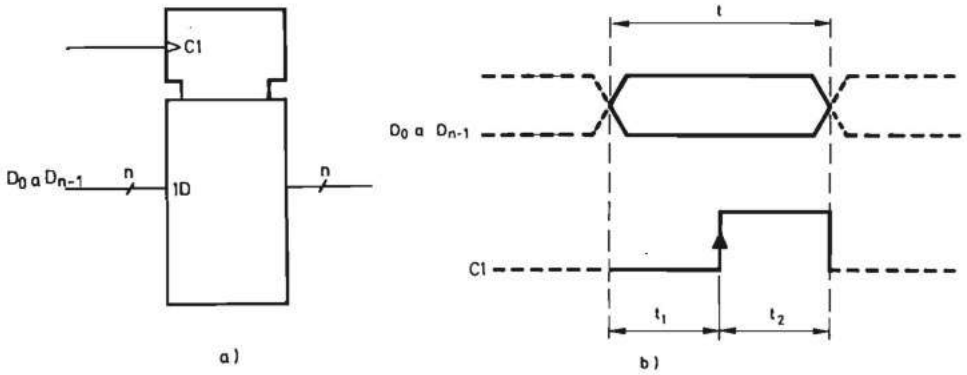


FIGURA 6.302.—**Ejemplo de especificación de la transferencia de información a un registro: *a)* Símbolo de un registro de entrada y salida en paralelo; *b)* Relación temporal entre las diferentes señales.**

representa un registro síncrono de entrada y salida en paralelo, en el cual la entrada T es activa con los flancos de subida. En la figura 6.302*b* s e indica que para que la transferencia se produzca es necesario que la información de las entradas D_0 a D_7 permanezca fija durante un intervalo t, y que en la entrada T se aplique un impulso cuyo flanco de subida aparezca al cabo de un tiempo t_1 desde que las citadas entradas se estabilizaron. También se indica que el estado de los bits D_0 a D_{n-1} es indiferente fuera del intervalo considerado.

La utilización de estas normas para la representación gráfica de los diagramas de evolución resulta especialmente interesante en el estudio de los procesadores digitales secuenciales síncronos que se estudian en el tomo II.

BIBLIOGRAFIA

[ALDA 80] F. Aldana y otros. Electrónica Industrial: Técnicas Digitales, Capítulo 12. Editorial Marcombo. 1980.

[HOLD 84] B. Holdsworth. Diseño de Lógica Digital. Capítulo 9. Editorial Gustavo Gili. 1984.

[INTE 88] Programmable Logic Handbook. Intel Corporation. 1988.

[INTER 81] Intersil. Data Book. Capítulo 6. Timers, counters and display drivers. 1981.

[MAND 83] E. Mandado. Sistemas Electrónicos Digitales. 4ª edición. págs. 316 a 322. Editorial Marcombo. 1983.

[MAND 87] E. Mandado. Sistemas Electrónicos Digitales. 6ª edición. Apartado 6.3.5.5, página 415. Editorial Marcombo. 1987.

[MAND 92] E. Mandado, J. Marcos, S.A. Pérez. Controladores Lógicos y Autómatas Programables. 2ª edición. Editorial Marcombo. 1992.

[MAND 95] E. Mandado, J. J. Rodríguez, L. J. Alvarez. Manual de prácticas de Electrónica Digital. Editorial Marcombo. 1995.

[MAXI 85] Maxim. Data Acquisition Catalog. Fixed and Programmable Time/Counters. pp. 6.53 a 6.60. 1985.

[MCCL 86] E.J. McCluskey. Logic Design Principles. Capítulo 7. Prentice-Hall International. 1986.

[MILL 84] J. Millman. Microelectrónica. Cap. 17. Generadores de Formas de Onda: Disparador de Schmitt. Editorial Hispano Europea. 1984.

[NATI 84*a*] National Semiconductor Corporation. Logic Data Book. CMOS oscillators. pp. 2.20 a 2.23. 1984.

[NATI 84*b*] National Semiconductor Corporation. Logic Data Book. HCMOS crystal oscillators. pp. 2.138 a 2.140. 1984.

[OSBO 81] Osborne 16 Bit Microprocessor Handbook. Introduction. Osborne/McGraw-Hill. 1981.

[PHIL 87] Semi-Custom Programmable Logic Devices (PLD). Philips data handbook. 1987.

[RONY 80] Interfacing Fundamentals: Timing Diagram Conventions. Computer Design. Enero. 1980.

[STON 83] H. S. Stone. Microcomputer Interfacing. Capítulo 5: Serial interfacing. Editorial Addison Wesley. 1983.

[TAUB 80] H. Taub, D. Schilling. Electrónica Digital Integrada. Cap. 2. Amplificadores Operacionales y Comparadores. Editorial Marcombo. 1980.

[TOCC 77] Tocci. Fundamentals of Pulse and Digital Circuits. C.E. Merril Publishing Company. 1977.

[UNGE 69] Unger, S.H. Asynchronous Sequential Switching Circuits. Wiley Interscience. 1969.

[UNGE 71] Unger, S.H. Asynchronous Sequential Switching Circuits with Unrestricted Input Changes. IEEE Trans. on Computers vol. C-20. págs. 1.437 a 1.444. Diciembre 1971.

[WAIT 85] Wait, Huelsman, Korn. Introducción al Amplificador Operacional. Teoría y Aplicaciones. Gustavo Gili, S.A. 1985.

Capítulo

7

Unidades de memoria

7.1 INTRODUCCION

Tal como se indica en el apartado 6.3.5.6.2 del capítulo 6, los procesadores digitales síncronos han de poseer una unidad de memoria en la que se almacenan los datos, los resultados parciales y los resultados finales del proceso. Cuando el sistema es especializado y el número de bits a memorizar es pequeño (no superior a 50), se puede utilizar registros de entrada y salida en paralelo o una serie disponibles como bloques funcionales de 4 a 8 bits en escala de integración media (MSI) en la mayoría de las tecnologías.

Cuando la cantidad de información a memorizar es elevada y, en especial, cuando el procesador digital es programable y han de memorizarse las instrucciones del programa además de los datos, nunca se necesita tener acceso a toda la información simultáneamente. Por ello los elementos que memorizan información se pueden agrupar constituyendo una unidad de memoria en la que sólo se puede acceder simultáneamente a un cierto número de aquéllos. Se define, por lo tanto, una unidad de memoria como un conjunto de biestables agrupados de tal manera que solamente es posible introducir información o leer la que poseen simultáneamente en un grupo reducido de ellos, que recibe el nombre de posición de la memoria.

UNIDAD DE MEMORIA

Conjunto de biestables agrupados en posiciones de tal manera que solamente es posible introducir información o leer la que poseen simultáneamente un número reducido de aquéllas.

Las características que ha de poseer una unidad de memoria dependen de la forma en que se va a presentar la información a su entrada, la forma en que se obtiene la información a su salida y la cantidad de información a memorizar.

En sucesivos apartados se hará referencia exclusivamente a las memorias que forman parte integrante de un procesador digital y que están físicamente situadas

en el mismo recinto físico que la unidad de control y la unidad operativa. Estas memorias constituyen lo que se denomina la memoria central del sistema o memorias auxiliares de la unidad aritmética. En primer lugar se indican los parámetros más importantes de una memoria y la división de las memorias de acuerdo con estos parámetros, y se analizan los diferentes tipos de memorias desde un punto de vista funcional independiente de la tecnología. Al final de este capítulo se exponen las tecnologías de uso más extendido.

7.2 PARAMETROS Y CARACTERISTICAS MAS IMPORTANTES DE UNA MEMORIA

7.2.1 Capacidad

Es la cantidad de información que puede almacenar. Dado que la información la almacena en el sistema binario, la capacidad se mide en la unidad de medida de información en dicho sistema, que es el bit. La capacidad se puede considerar, por lo tanto, igual al número N total de bits. Pero, en general, la información se almacena en grupos de bits denominados posiciones o palabras formadas por un cierto número n_1 de bits accesibles simultáneamente. El número de palabras o posiciones de una memoria viene dado, por lo tanto, por el cociente $N/n_1 = m$.

A la selección de una determinada posición de memoria para introducir información en ella o leer la que contiene se la denomina direccionamiento de dicha posición. Para que se aproveche al máximo la capacidad de selección se hace que m sea una potencia de dos. El número de variables binarias necesarias para poder seleccionar m posiciones es n_2 tal que $2^{n_2} = m$. La unidad de capacidad en lo que se refiere al número de posiciones de memoria se considera igual a 1024 (por ser la potencia de dos más próxima a 1.000) y se la denomina k. Para seleccionar 1024 posiciones se necesitan 10 bits, dado que $2^{10} = 1024$. De todo lo expresado se deduce que la capacidad en k de una memoria seleccionada mediante n_2 variables es 2^{n_2-10}. Como práctica el lector puede comprobar que si $n_2 = 14$, la memoria tiene una capacidad de 16 k.

El número total de posiciones de una memoria de n k es $n \cdot 1024$. Por ejemplo una memoria de 32 k posee $32 \cdot 1024 = 32798$ posiciones.

La capacidad total de una memoria se suele indicar mediante el número de posiciones y el número de bits de cada posición. Por ejemplo, una memoria de 32 k 16 posee 32.798 posiciones de 16 bits cada una y su capacidad total es $N = 32.798 \cdot 16 = 524.768$ bits. Es también usual que la capacidad de una memoria se mida en octetos (bytes) que son grupos de 8 bits. Cuando se hace así una memoria de 512 k por ejemplo posee $512 \cdot 1024 \cdot 8 = 2.146.304$ bits.

7.2.2 Forma de acceder a la información

Es éste un parámetro de gran importancia. La acción de introducir información (escribir) en una posición de la memoria o leer la que contiene se denomina acceder a la misma. Tal como se dijo en la introducción, nunca es necesario escribir o leer

simultáneamente todas las posiciones de una memoria. Por ello existen varias formas de acceder a una posición de una memoria que dan lugar a la división de las memorias en dos grandes clases, indicadas en la parte izquierda de la figura 7.1:

a) Memorias de acceso directo.

Son aquellas en las que es posible especificar una posición o conjunto de posiciones para leer su información o escribir en ella.

A la acción de especificar una posición o conjunto de posiciones se la denomina direccionamiento que se realiza mediante una combinación binaria tal como se indica en el apartado anterior.

b) Memorias de acceso secuencial.

Son aquellas en las que no se especifica la dirección de la información sino que en cada instante se introduce en la memoria la información situada en los terminales de entrada o bien se lee la situada en la posición unida a los terminales de salida.

Las memorias de acceso directo y de acceso secuencial se pueden realizar de múltiples maneras. El desarrollo de la microelectrónica, que ha permitido la realización de estructuras complejas con un coste reducido, ha hecho que las unidades de memoria hayan tenido una evolución muy rápida y que algunas estructuras hayan perdido interés práctico.

En sucesivos apartados se analizan las estructuras internas más importantes de las unidades de memoria y los circuitos combinados con ellas que permiten realizar memorias de acceso directo y de acceso secuencial.

7.2.3 Estructura interna

La estructura interna de una memoria se define como la forma en que se conectan entre sí las diferentes posiciones.

En la parte derecha de la figura 7.1 se representa la clasificación de las unidades de memoria de acuerdo con su estructura interna. En letra minúscula sin recuadrar se indican las características o parámetros importantes, y en letra mayúscula en el interior de un recuadro los diversos tipos de memorias a que da lugar cada una de ellas.

En esta clasificación se incluyen solamente, tal como se indicó anteriormente, las memorias que no poseen partes en movimiento y que, en general, se encuentran en el mismo recinto físico que la unidad de control y la unidad aritmética. El almacenamiento de información en cantidades elevadas (cientos de miles de k palabras), tanto de datos como de instrucciones de los sistemas programables, se realiza actualmente en sistemas físicos que poseen partes en movimiento y reciben el nombre genérico de periféricos.

En sucesivos apartados se estudian los diferentes tipos de memorias.

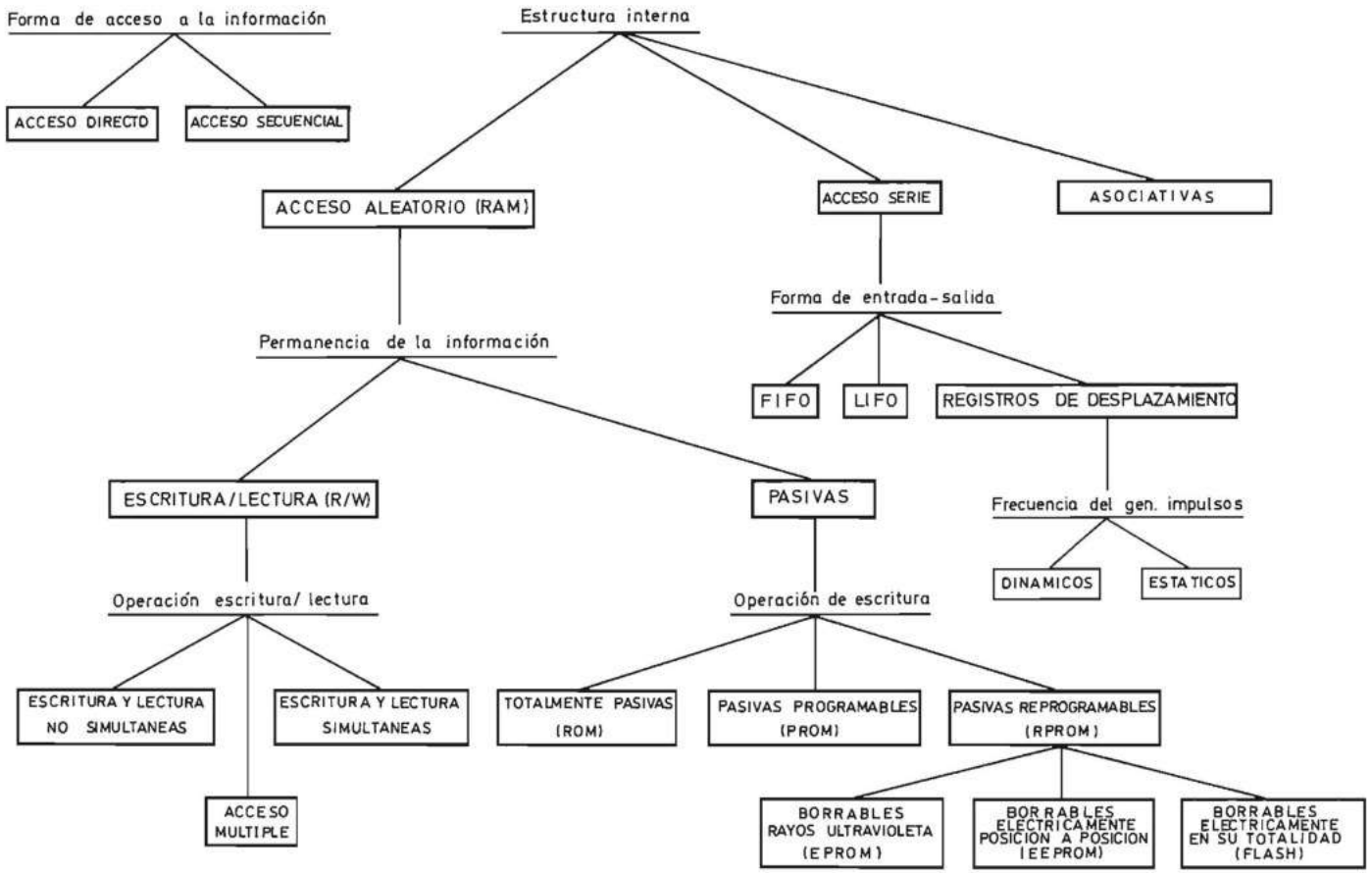


FIGURA 7.1.—**Clasificación de las memorias.**

7.2.3.1 Memorias de acceso aleatorio (Random Access Memories)

7.2.3.1.1 Características generales. Son memorias que poseen un conjunto de variables de dirección que permiten seleccionar cualquier posición de la misma. Se caracterizan porque el tiempo que tarda en leerse o grabarse una posición, a partir del instante en que se presenta la dirección, es el mismo para todas las posiciones de la memoria. Por lo tanto, una memoria de acceso aleatorio es por sí misma una memoria de acceso directo en la que el tiempo de acceso a cualquier posición es el mismo independientemente de su situación.

MEMORIA DE ACCESO ALEATORIO

Conjunto de biestables agrupados en posiciones seleccionables mediante un conjunto de variables de dirección, que presenta la característica de que el tiempo que tarda en leerse o escribirse en cualquier posición es el mismo independientemente de su situación.

En el caso de que solamente se pueda seleccionar simultáneamente una posición de la memoria para leer su contenido o introducir información en ella, el símbolo lógico de una memoria de acceso aleatorio es el representado en la figura 7.2.

En ella se supone que la memoria está organizada en posiciones de n_1 bits. Por ello el número de terminales de entrada y salida de información es n_1. La selección de la posición cuya información se desea leer o escribir se realiza mediante n_2 bits en paralelo, denominados bits de dirección (fig. 7.2). El número total de

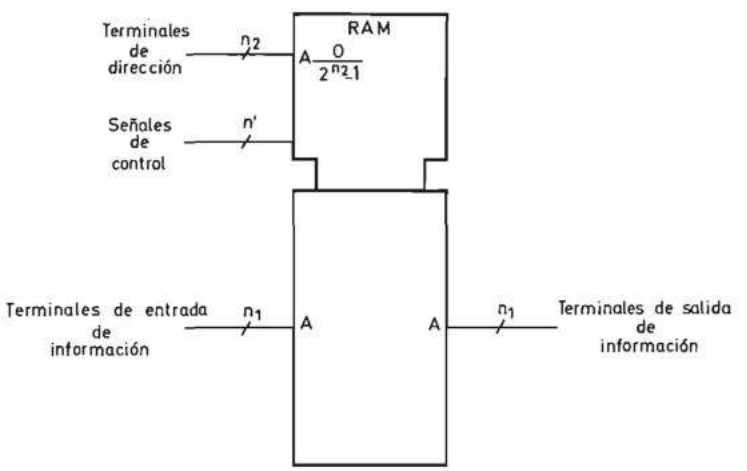


FIGURA 7.2.—**Símbolo lógico de una memoria de acceso aleatorio de escritura y lectura no simultáneas con terminales de entrada y salida independientes.**

posiciones que se pueden seleccionar es 2^{n_2}. Por ejemplo, una memoria de 32 k tiene una dirección constituida por 15 bits ($2^{15} = 32.768 = 32k$).

En la figura 7.2 se utiliza un símbolo normalizado. Las variables de dirección (address) llevan el indicativo $A \frac{0}{2^{n_2}-1}$ para especificar que seleccionan las 2^{n_2} posiciones. Los terminales de entrada se indican con la letra A para especificar que la información presente en ellos se introduce en una operación de escritura en la posición seleccionada mediante las variables de dirección. De forma similar los terminales de salida se indican también con la letra A.

Además de los terminales de entrada y salida de información y de dirección, una memoria de acceso aleatorio posee un conjunto n' de señales de control que tienen como finalidad gobernar la realización de las acciones de escritura y lectura. Los tipos de señales de control dependen de la forma en que está realizada la memoria y de sus aplicaciones y, por otra parte, no existe uniformidad en las denominaciones de los diversos fabricantes. Por ello antes de estudiar las señales de control es conveniente anàlizar los conceptos más importantes de una memoria de acceso aleatorio que son el tiempo de acceso y el tiempo de ciclo.

El tiempo de acceso es el que transcurre desde que se da una orden de lectura hasta que la información contenida en la posición seleccionada aparece a la salida, y se representa de forma gráfica en la figura 7.3. A partir del instante en que las variables de dirección permanecen estables se inicia la lectura de la posición correspondiente colocando las señales de control en los niveles adecuados. Al cabo de un tiempo, denominado t_{acceso}, aparece en los terminales de salida la información contenida en la posición seleccionada, si se produce la situación que se acaba de describir.

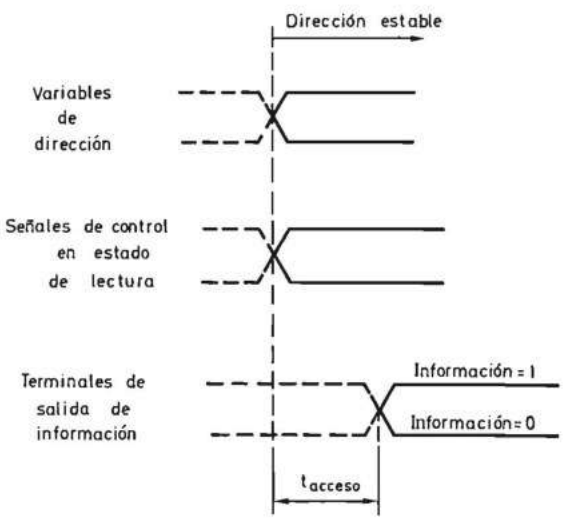


FIGURA 7.3.—**Tiempo de acceso de una memoria de acceso aleatorio.**

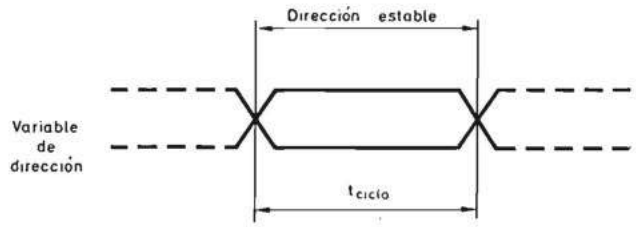


FIGURA 7.4.—**Tiempo de ciclo de una memoria de acceso aleatorio.**

El tiempo de ciclo es el que ha de transcurrir como mínimo entre dos órdenes de escritura o lectura. En general la operación de escritura necesita más tiempo que la de lectura y es la que condiciona el tiempo de ciclo. En la figura 7.4 se representa el tiempo de ciclo de forma gráfica indicando que durante él, la dirección ha de permanecer estable. En la figura 7.5 se representa la relación temporal entre las señales que hay que aplicar a una memoria de acceso aleatorio para que se realice una operación de escritura. Durante un cierto intervalo de tiempo denominado $t_{escritura}$ se ha de colocar en los terminales de dirección la combinación binaria correspondiente a la posición de la memoria en la que se va a introducir la informa-

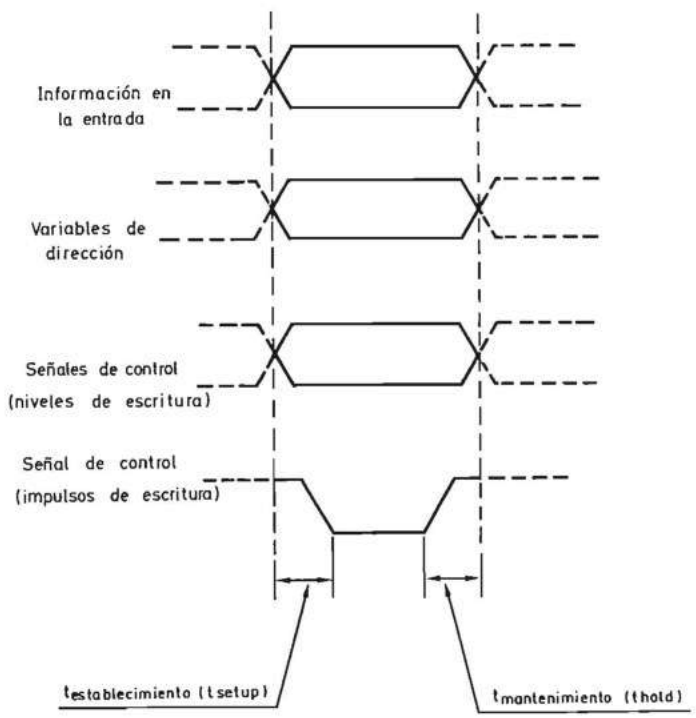


FIGURA 7.5.—**Señales de una memoria de acceso aleatorio durante un ciclo de escritura.**

ción presente en los terminales de entrada. Para ello las señales de control han de permanecer en los niveles de escritura, y en general, una de ellas ha de ser un impulso de escritura activo con un uno o con un cero lógicos (en la figura 7.5 se supone activo con un cero lógico) y que ha de alcanzar ese nivel un cierto tiempo llamado de establecimiento $t_e(t_{set\ up})$ después de que la dirección está estable y abandonarlo un cierto tiempo, llamado de mantenimiento t_m (t_{hold}), antes de que la dirección vuelva a cambiar.

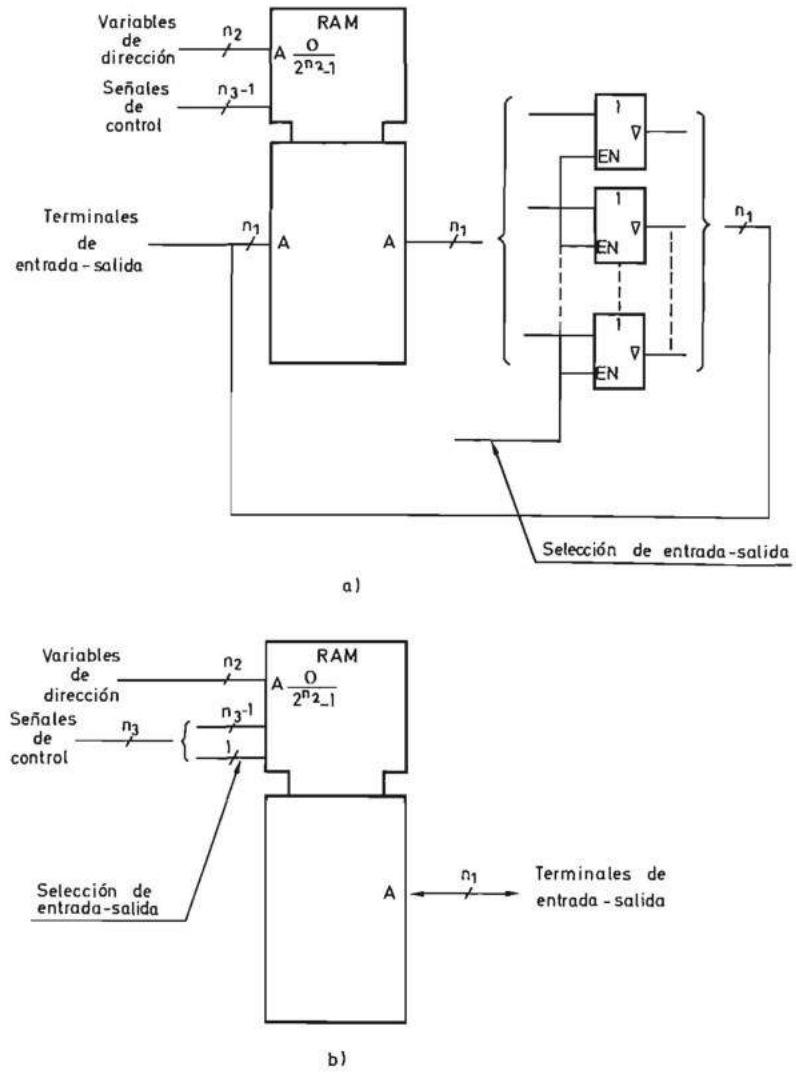


FIGURA 7.6.—**Memoria de acceso aleatorio con terminales comunes de entrada y salida: *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

En la figura 7.2 se ha supuesto que la memoria posee terminales de entrada y salida independientes. Pero es usual que, si sólo se puede seleccionar una posición de memoria y escribir en ella o leerla, ambas acciones se realicen en instantes diferentes de tiempo a través de unos terminales únicos. Para ello los terminales de salida se unen con los de entrada a través de puertas de tres estados y entre las señales de control debe existir una que, de forma exclusiva o combinada con otras acciones, controle el tercer estado de las mismas. En la figura 7.6*a* se representa la memoria con los amplificadores indicados y en la figura 7.6*b* el símbolo lógico equivalente.

Por otra parte es interesante en muchas aplicaciones que la memoria posea salidas de tres estados que permitan su conexión a una barra (bus) de información. Para ello, entre las señales de control se puede incluir una que se denomina desinhibición de salida (output enable) que actúa sobre la entrada de desinhibición (EN) del amplificador de tres estados de la figura 7.6*a*. En la figura 7.7*a* se representa el esquema de una memoria de acceso aleatorio con salida de tres estados, que se diferencia del de la figura 7.6*a* en la existencia de una puerta Y (AND) a la que además del terminal de selección de entrada/salida, se conecta un terminal de desinhibición de salida. En la figura 7.7*b* se indica el símbolo lógico correspondiente.

7.2.3.1.2 Organización de las memorias de acceso aleatorio. Tal como se indicó en el apartado anterior, las memorias de acceso aleatorio están formadas por células biestables organizadas en posiciones de n_1 células cada una. Para seleccionar la posición en la que se introduce o de la que se lee la información se utiliza un decodificador, que tal como se vio en el capítulo 3 es un sistema combinacional que tiene en general un número de salidas igual al de combinaciones posibles de las variables de entrada, de tal forma que para cada combinación de las variables de entrada se activa una y sólo una variable de salida (ver apartado 3.7.1).

En la figura 7.8 se representa el diagrama de bloques de una memoria de acceso aleatorio.

Según la forma de seleccionar las posiciones de una memoria de acceso aleatorio se distinguen tres estructuras u organizaciones que se denominan 2 D, 3 D y 2½ D y se estudian a continuación.

a) Organización 2 D, por palabras o lineal.

Se denomina así porque existen tantos terminales de unión diferentes entre el decodificador y las posiciones de memoria como número de éstas posee la memoria. En la figura 7.9 se representa el diagrama de bloques de una memoria 2 D de 2^{n_2} posiciones de n_1 bits cada una. Las variables de dirección se decodifican mediante un único decodificador de 2^{n_2} salidas, cada una de las cuales se conecta a una de las 2^{n_2} posiciones de la memoria. Los terminales de entrada y salida de todas las posiciones se conectan entre sí, al igual que la señal que selecciona la escritura o lectura de la memoria. Al aparecer en las entradas de las variables de dirección una determinada combinación binaria se activa una sola salida del decodificador y las células de la posición conectada a ella son leídas o escritas según el estado de la señal de control de escritura/lectura.

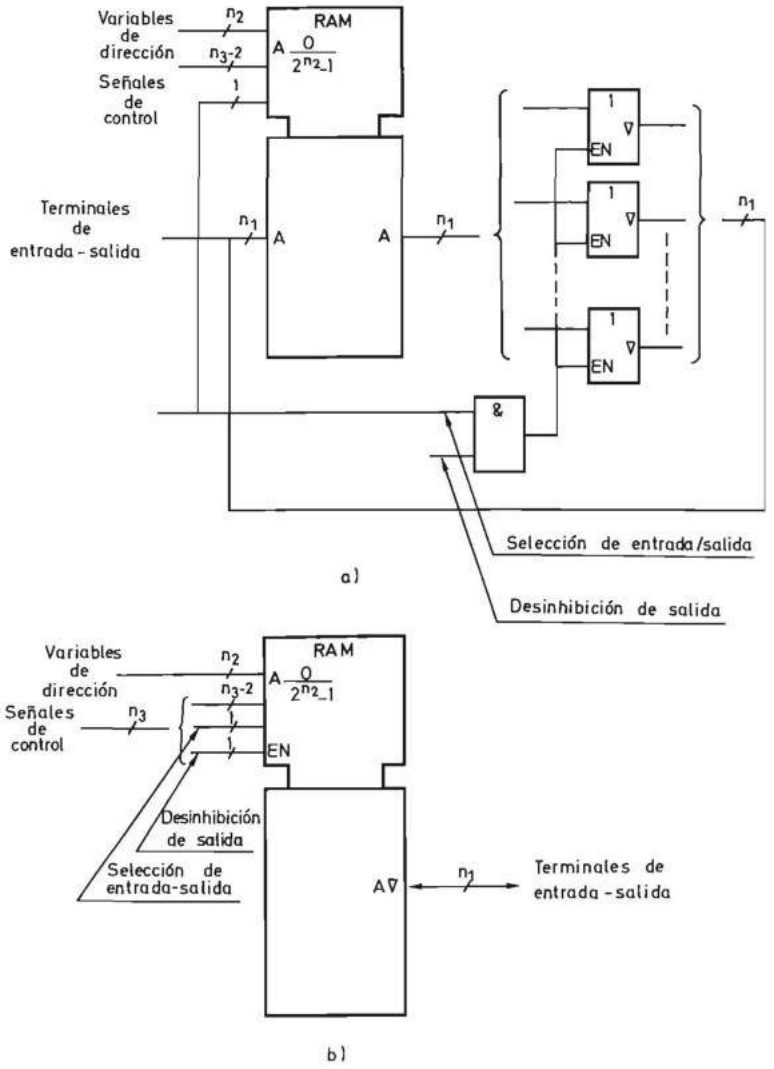


FIGURA 7.7.—**Memoria de acceso aleatorio con terminales comunes de entrada y salida, y salida de tres estados: *a*) Esquema del circuito; *b*) Símbolo lógico normalizado.**

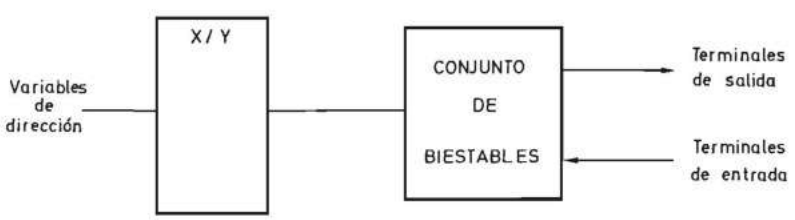


FIGURA 7.8.—**Diagrama de bloques de una memoria de acceso aleatorio.**

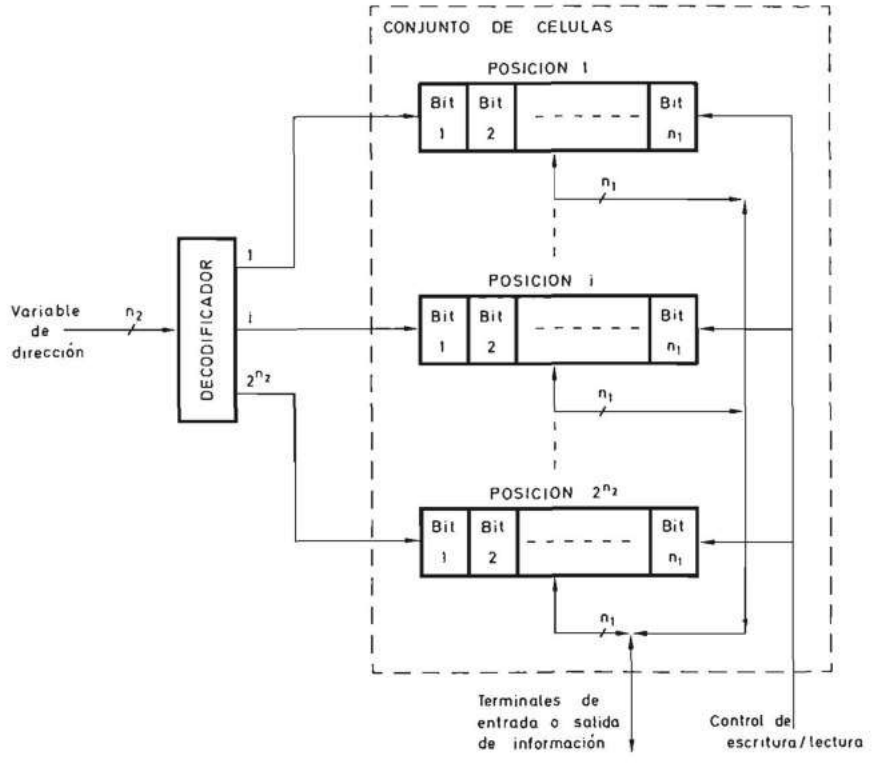


FIGURA 7.9.—**Organización 2D de una memoria de acceso aleatorio.**

El número de conexiones entre el conjunto de posiciones y el exterior a ellas es:

$$N_c = 2^{n_2} + n_1 + 1$$

lo cual puede comprobar fácilmente el lector en la figura 7.9.

Un inconveniente de esta organización es que el número de conexiones entre el conjunto de células y el exterior a ellas crece linealmente con el número de posiciones. Por ejemplo, el número de conexiones de una memoria de 4 k 8 para la cual se necesitan 12 variables de selección (2^{12} = 4 k) es:

$$N_c = 2^{12} + 8 + 1 = 4096 + 8 + 1 = 4105$$

Se observa que el número de conexiones de entrada y salida y de control es despreciable con respecto al de conexiones de selección de posición.

b) Organización 3 D o por coincidencia.

Esta organización reduce el número de conexiones entre las posiciones de memoria y el exterior, dividiendo las variables de dirección en dos grupos que se de-

codifican por separado y seleccionando las posiciones mediante dos salidas, una de cada uno de los decodificadores (fig. 7.10). A continuación se demuestra que el número mínimo de conexiones se obtiene cuando ambos decodificadores son iguales, es decir, cada uno decodifica la mitad de las variables de dirección.

En efecto, denominando X al número de salidas de un decodificador e Y al del otro, el número de conexiones de las posiciones de la memoria con el exterior es:

$$N_c = X + Y + n_1 + 1$$

Pero siendo 2^{n_2} el número total de posiciones se ha de verificar

$$X \cdot Y = 2^{n_2}$$

de donde

$$Y = \frac{2^{n_2}}{X}$$

y sustituyendo en la expresión de N_c resulta:

$$N_c = X + \frac{2^{n_2}}{X} + n_1 + 1$$

Para obtener el valor de X que hace mínimo a N_c calcularemos la derivada de N_c con respecto a X y la igualaremos a cero:

$$\frac{dN_c}{dX} = 1 + 2^{n_2}\left(-\frac{1}{X^2}\right) = 0$$

$$-\frac{2^{n_2}}{X^2} = -1$$

$$X = \sqrt{2^{n_2}} = 2^{n_2/2}$$

y por lo tanto resulta:

$$Y = \frac{2^{n_2}}{X} = \frac{2^{n_2}}{2^{n_2/2}} = 2^{n_2/2} = X$$

Esta organización reduce enormemente el número de conexiones diferentes entre las células de memoria y el exterior con respecto a las 2 D cuando el número de posiciones es elevado y, por ello, es la más utilizada por los fabricantes de circuitos integrados para la realización de memorias de acceso aleatorio.

Como ejemplo calcularemos el número de conexiones necesarias en una memoria de 4 k 8 realizadas en organización 3 D. El número de variables de dirección necesarias para seleccionar 4 k es 12 ($2^{12} = 4$ k) y, por lo tanto, resulta:

$$N_c = 2^{12/2} + 2^{12/2} + 8 + 1 = 2 \cdot 2^6 + 8 + 1 = 137$$

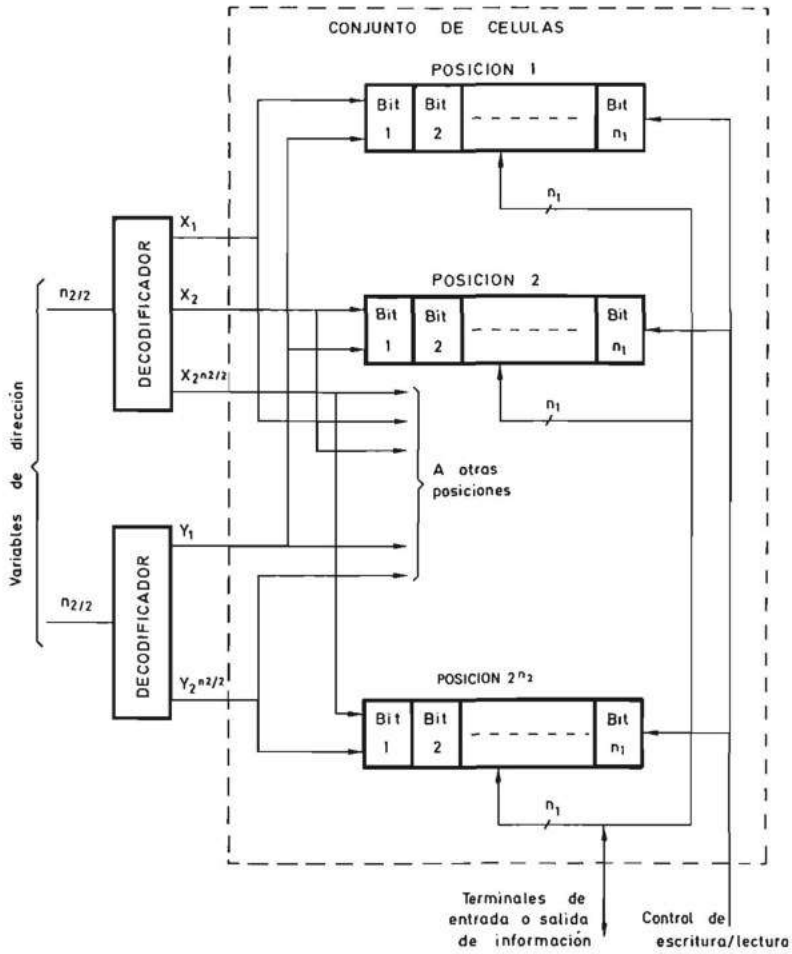


FIGURA 7.10.—**Organización 3D de una memoria de acceso aleatorio.**

En la figura 7.10 se representa el diagrama de bloques de la organización 3 D. Se observa que cada salida de los decodificadores se conecta a $2^{n_2/2}$ posiciones.

c) Organización 2½ D.

Este tipo de organización, constituye una alternativa a la 3 D y trata de reducir el número de conexiones de selección que llegan a las posiciones, sustituyendo uno de los dos decodificadores de la figura 7.10 por el conjunto formado por un multiplexor y un demultiplexor.

En la figura 7.11 se representa el esquema correspondiente a la organización 2½ D.

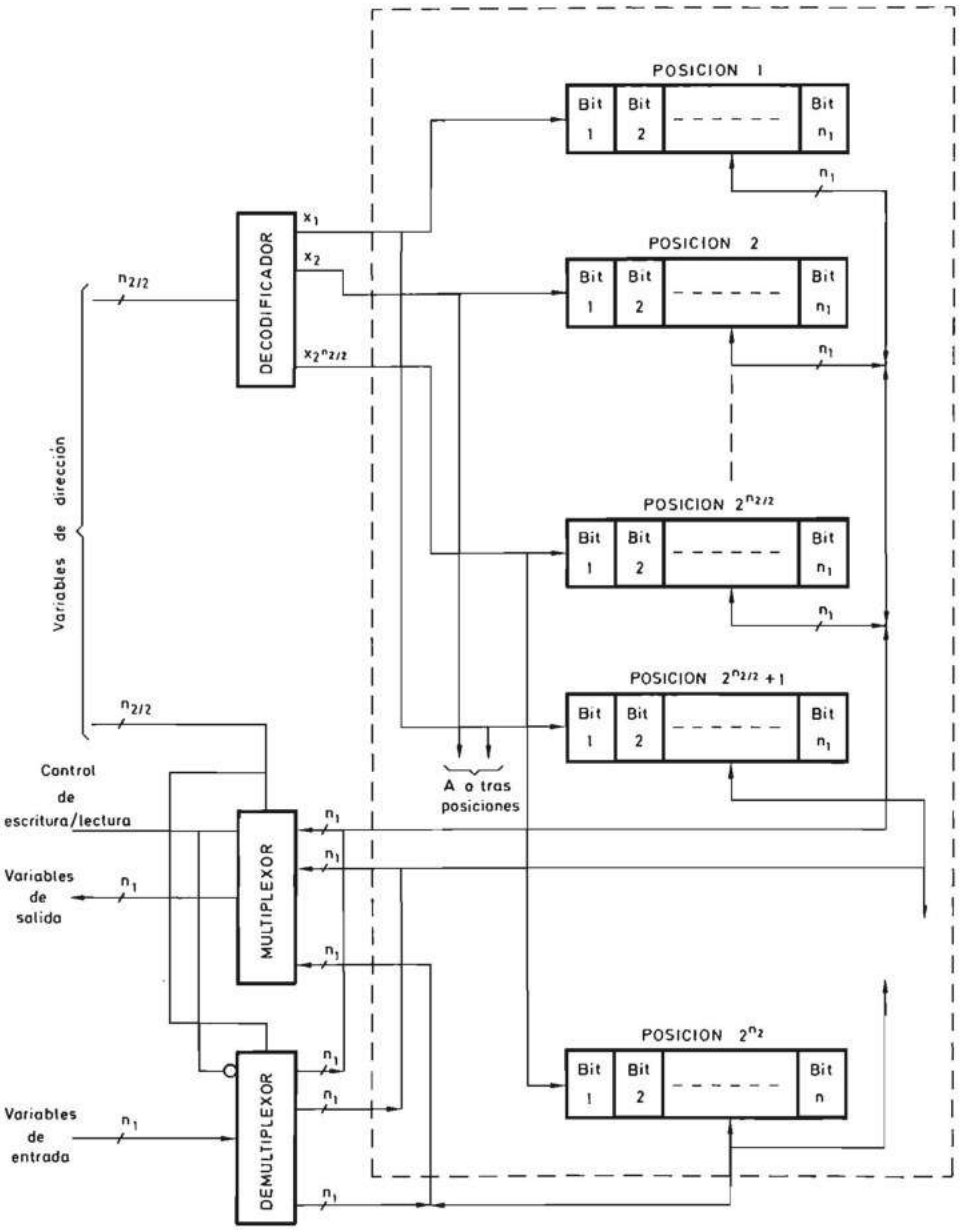


FIGURA 7.11.—**Organización 2 ½ D de una memoria de acceso aleatorio.**

La mitad de las variables de dirección se conectan a un decodificador cuyas salidas se unen a las diferentes posiciones al igual que en la organización 3 D. Por lo tanto, cada salida X de este decodificador se conecta a $2^{n_1/2}$ del total de posi-

ciones (el producto de $2^{n_2/2}$ salidas del decodificador por $2^{n_2/2}$ posiciones a las que se conecta cada una da el total de posiciones que es 2^{n_1}). Se forman $2^{n_2/2}$ grupos de posiciones en los cuales existe una posición en cada uno conectada a una de las salidas del decodificador y se conectan entre sí los terminales de entrada/salida de todas las posiciones de cada grupo. Esta conexión común se une a una de las entradas de un multiplexor y una de las salidas de un demultiplexor de $2^{n_2/2}$ canales seleccionados mediante las restantes $2^{n_2/2}$ variables de dirección.

El demultiplexor se utiliza en la operación de escritura. Sus entradas reciben la información a escribir que va a parar a la salida seleccionada mediante las variables conectadas a las entradas de selección y se introduce en la posición seleccionada por una variable x_i del decodificador.

El multiplexor se utiliza en la operación de lectura. Sus $2^{n_2/2}$ entradas se conectan a cada uno de los grupos de posiciones. A su salida aparece la información seleccionada mediante las $2^{n_2/2}$ variables de selección conectadas a las entradas correspondientes del multiplexor y la salida x_i del decodificador.

La señal de control de escritura/lectura desinhibe el multiplexor en la operación de lectura y el demultiplexor en la de escritura.

De lo expuesto se deduce que existe una alternativa clara entre la organización 3 D y $2\frac{1}{2}$ D que se decanta a favor de esta última cuando el número de posiciones de la memoria es muy elevado.

7.2.3.1.3 Permanencia de la información. Este parámetro se mide de forma cualitativa por la diferencia entre el tiempo de lectura y el tiempo de escritura.

De acuerdo con él, las memorias de acceso aleatorio se clasifican en memorias de escritura/lectura (R/W) o activas y memorias pasivas tal como se indica en la figura 7.1. A continuación se estudia cada uno de ellas.

7.2.3.1.3.1. Memorias de escritura/lectura (R/W) o activas. Son aquellas memorias de acceso aleatorio en las que los tiempos necesarios para realizar una operación de escritura o lectura son del mismo orden de magnitud. En general los fabricantes de circuitos integrados suelen dar a estas memorias la denominación de memorias de acceso aleatorio (RAM) lo cual no es muy apropiado, porque existen memorias de acceso aleatorio que se estudian en sucesivos apartados, cuyo comportamiento es diferente.

Las memorias de acceso aleatorio activas se realizaron inicialmente con ferritas, que a partir de 1970 fueron desplazadas por las tecnologías de semiconductores, algunas de las cuales se describen en el apartado 7.3.

Las memorias de acceso aleatorio activas se pueden clasificar de acuerdo con diversas características que dan lugar a un conjunto de tipos que se interrelacionan entre sí y que se indican en la tabla 7.1.

A continuación se analiza cada una de estas características.

a) Duración de la memorización.

Se define como duración el tiempo máximo que transcurre desde que se intro-

TABLA 7.1.— **Clasificación de las memorias de acceso aleatorio activas.**

duce la información en la memoria hasta que ésta se pierde, en el supuesto de que la tensión de alimentación se mantenga de forma indefinida.

Se denominan estáticas (SRAM) las memorias que mantienen indefinidamente la información mientras se les aplica la tensión de alimentación. Las memorias estáticas de semiconductores están formadas por biestables activados por niveles en general del tipo R-S como los representados en las figuras 6.21 y 6.22. En su realización se pueden utilizar transistores bipolares o transistores MOS.

Se denominan dinámicas (DRAM) las memorias que mantienen almacenada la información solamente durante algunos milisegundos. Estas memorias sólo se realizan con transistores MOS y en lugar de biestables, utilizan como elemento de memoria la capacidad parásita existente entre la puerta y el surtidor de aquéllos. Para que la información se memorice de forma indefinida es necesario realizar una lectura y escritura periódicas que reciben el nombre de regrabado (refreshing).

Para un estudio detallado de la realización física de las memorias estáticas y dinámicas se remite al lector al apartado 7.3.2.1

b) Volatilidad de la información.

La volatilidad es la propiedad que tiene una memoria de perder la información almacenada al desaparecer la tensión de alimentación. Se dice que son volátiles las memorias en las que se produce dicha desaparición y que no son volátiles en caso contrario. Las memorias de ferrita no son volátiles y las de semiconductores activas lo son.

c) Forma de realizar las operaciones de escritura y lectura.

De acuerdo con esta característica las memorias de acceso aleatorio activas pueden ser de tres tipos:

7.2.3.1.3.1a Memorias activas de escritura y lectura no simultáneas Este tipo de memorias es el que suele recibir el nombre de RAM por parte de los fabricantes de circuitos integrados. En él sólo se puede seleccionar en cada instante una posición de memoria para realizar la operación de lectura o escrita en ella. Su diagrama de bloques es idéntico al representado en las figuras 7.2 y 7.6.

Las memorias activas de escritura y lectura no simultáneas pueden ser estáticas (SRAM) o dinámicas (DRAM).

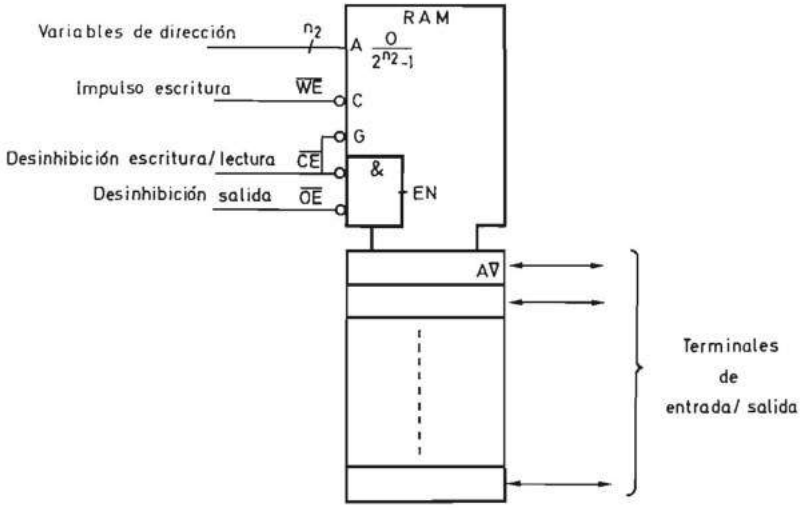


FIGURA 7.12.—**Símbolo lógico de una memoria de acceso aleatorio estática de escritura y lectura no simultáneas.**

No existe uniformidad en el número de terminales de control de las memorias de acceso aleatorio estáticas pero las más versátiles tienen tres terminales que corresponden a otras tantas señales de control que pueden ser activas con el nivel cero o el nivel uno. Estas señales se indican en la figura 7.12 y se describen a continuación:

— Desinhibición de escritura/lectura [Chip Enable (CE) o Chip select (CS)]. En la figura 7.12 se supone activa con un cero lógico ($\overline{CE}$). Cuando esta señal adopta el nivel activo permite que se realice una operación de escritura o de lectura y en caso contrario lo impide. En general suele además reducir la potencia consumida por el circuito cuando está desactivada.

— Desinhibición de salida [Output enable (OE)]. En la figura 7.12 se supone activa con un cero lógico ($\overline{OE}$). Cuando está desactivada pone en tercer estado la salida. Por el contrario si se encuentra en nivel activo, hace que en la salida aparezca la información de la posición de memoria seleccionada, si la desinhibición de escritura/lectura (CE) se encuentra también en el nivel activo.

— Impulso de escritura [Write enable [(WE)]. Recibe este nombre porque consiste en un impulso que si aparece durante el intervalo de tiempo en que las variables de dirección permanecen estables, produce la escritura en la posición seleccionada de la información presente en los terminales de entrada/salida. Para que dicha acción tenga lugar es necesario que la desinhibición de escritura/lectura (CE) esté también activada.

En las figuras 7.13 y 7.14 se indican las relaciones temporales entre las distintas señales de la memoria de la figura 7.12 para que se realice una lectura y una escritura respectivamente.

Una de las principales aplicaciones de las memorias activas de escritura y lectura no simultáneas es la de constituir la memoria de datos y/o de instrucciones de un procesador digital que se estudia en el tomo II. En particular, si la unidad

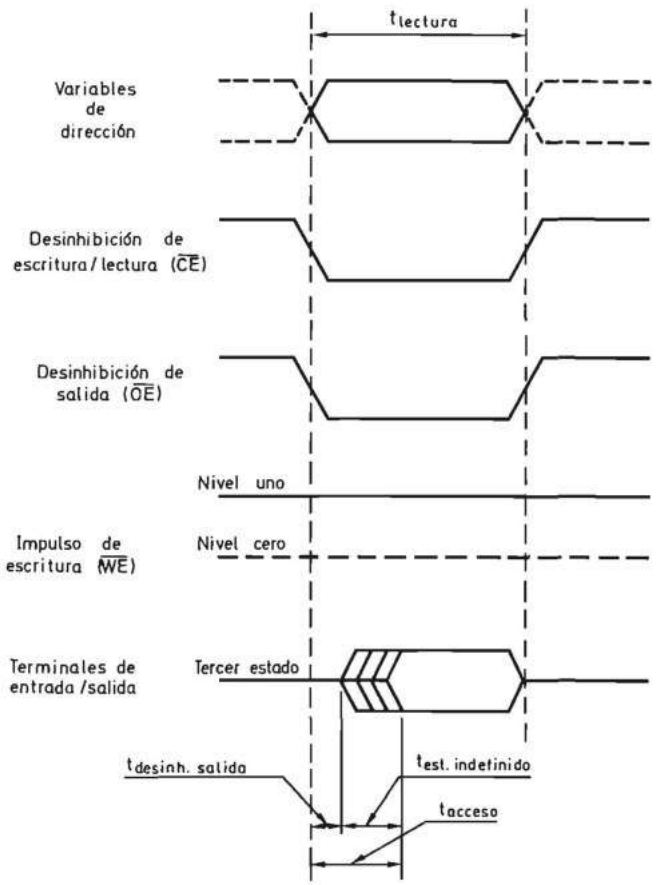


FIGURA 7.13.—**Relación temporal entre las señales aplicadas a una memoria de acceso aleatorio estática durante un ciclo de lectura.**

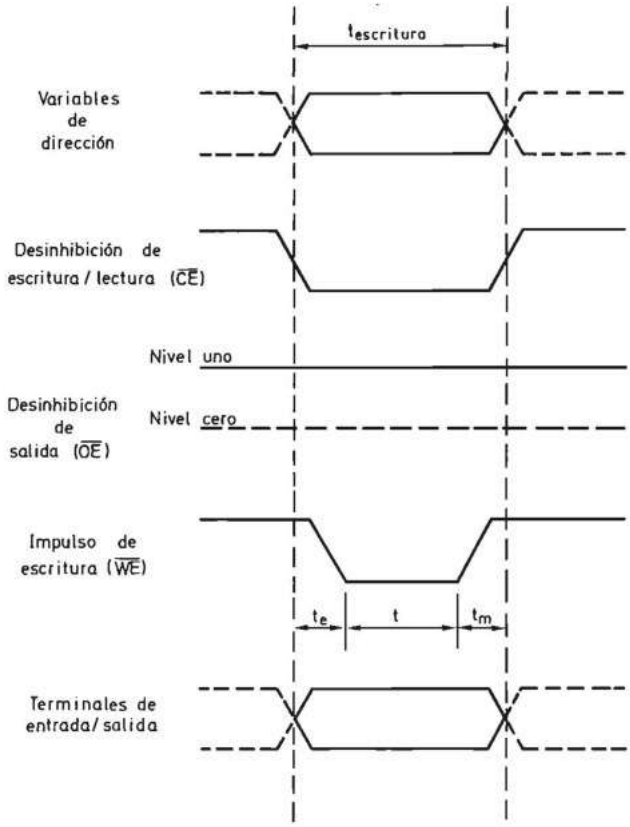


FIGURA 7.14.—**Relación temporal entre las señales aplicadas a una memoria de acceso aleatorio estática durante un ciclo de escritura.**

central de proceso posee terminales independientes para los datos, las direcciones y las señales de control, el esquema correspondiente es el representado en la figura 7.15.

Las memorias dinámicas se pueden realizar de muy diferentes maneras. Por ello a continuación se analizan los conceptos comunes a todas ellas para que el lector pueda interpretar las hojas de características de los diferentes tipos [INTE 86a] [NEC 87] [TEXA 89]

Las memorias de acceso aleatorio dinámicas han tenido un desarrollo espectacular desde que se realizó la primera que fue la 1103 de tecnología MOS canal P de Intel que poseía una capacidad de 1K1. Las memorias dinámicas son en la actualidad de MOS canal N o CMOS. La organización interna suele ser 3 D y poseen un amplificador de regrabado por columna (ver apartado 7.3.2.1). Por ello su diagrama de bloques es el representado en la figura 7.16 en el cual los n_2 bits de dirección o selección se dividen en dos grupos de $n_2/2$ bits cada uno. Para poder integrar memorias de gran capacidad, de hasta 1Mbit en único circuito integrado, las memorias dinámicas suelen poseer registros temporales para memorizar la dirección. Es-

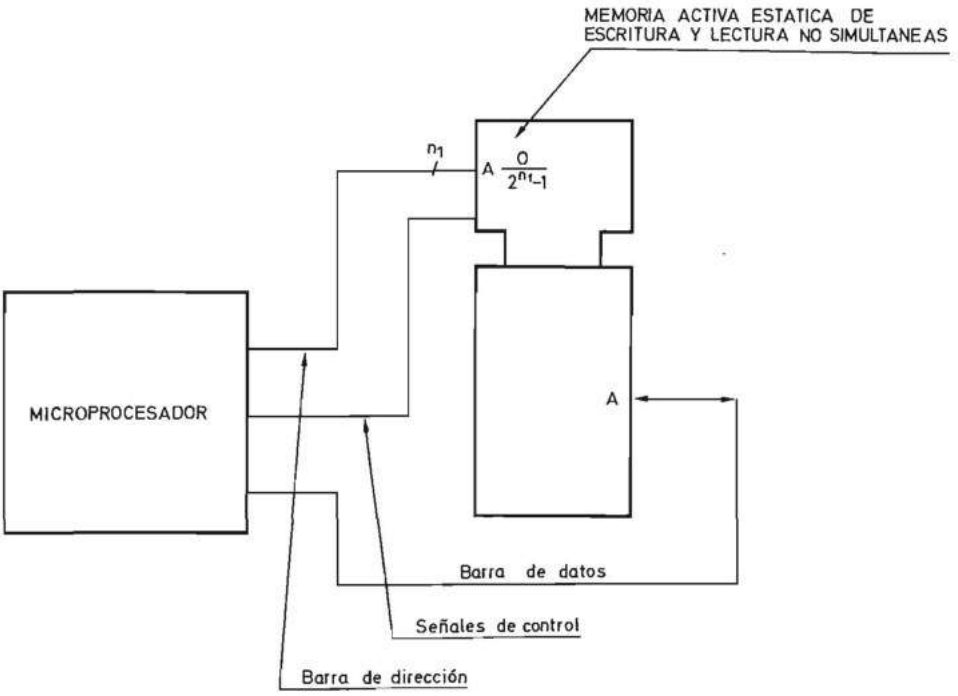


FIGURA 7.15.—**Memoria de acceso aleatorio estática de escritura y lectura no simultáneas utilizada como unidad de memoria de un procesador digital.**

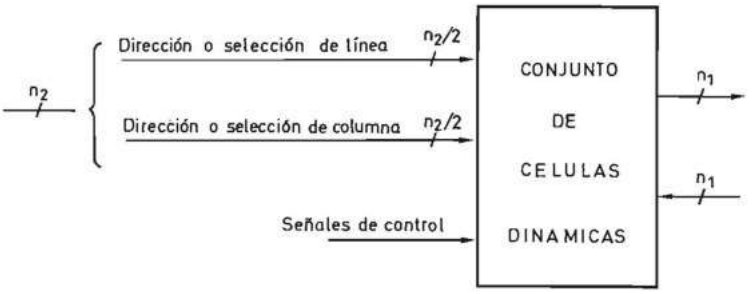


FIGURA 7.16.—**Diagrama de bloques de una memoria de acceso aleatorio dinámica.**

tos registros poseen entradas de información comunes tal como se representa en la figura 7.17 para reducir el número de terminales necesarios. El registro de la dirección de líneas suele ser activado por flancos mediante una señal que recibe el nombre de «Muestreo de la dirección de las líneas» [Row address strobe (RAS)]. En la figura 7.17 esta señal es activa con los flancos de bajada y por ello se denomina $\overline{\text{RAS}}$. De forma similar la señal $\overline{\text{CAS}}$, (Column address strobe) también

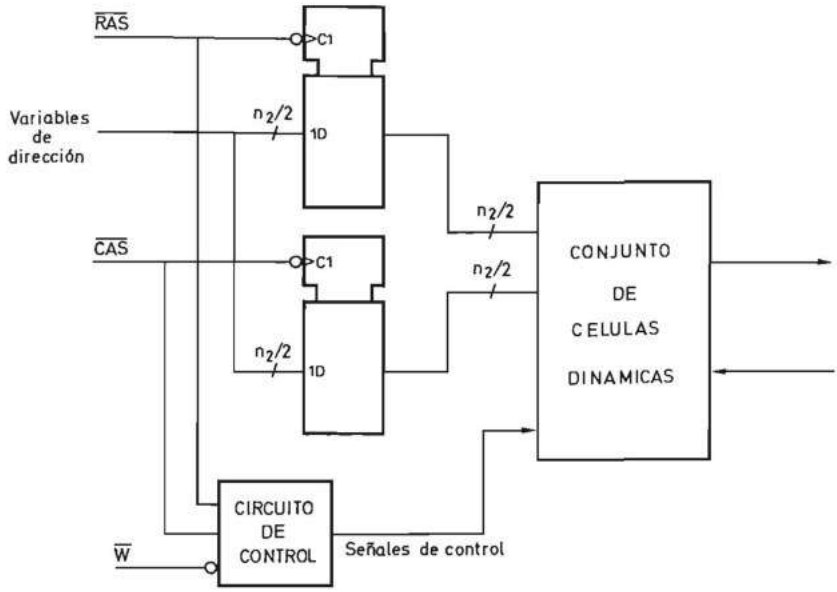


FIGURA 7.17.—**Diagrama de bloques de una memoria de acceso aleatorio dinámica con registros temporales de memorización de la dirección.**

activa con flancos de bajada, produce la memorización de la dirección de las columnas. Además, la memoria posee una señal del modo de operación (escritura o lectura) W. En la figura 7.17 recibe la denominación $\overline{W}$ porque cuando se encuentra en nivel cero se produce una escritura (write) y cuando está en uno, una lectura (read). Las señales $\overline{RAS}$, $\overline{CAS}$ y $\overline{W}$ se aplican a un circuito que genera las señales de control de las células de memoria. Se obtiene así el diagrama de bloques de una memoria dinámica representado en la figura 7.18.

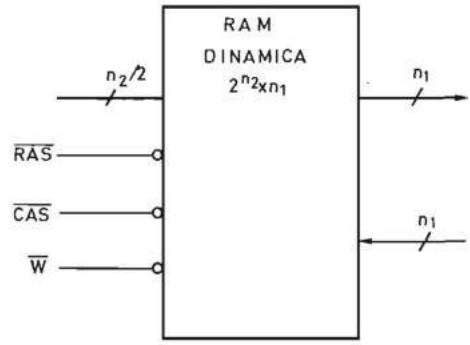


FIGURA 7.18.—**Símbolo lógico de una memoria de acceso aleatorio dinámica con registros temporales de memorización de la dirección.**

Pero las memorias dinámicas necesitan, para no perder la información que poseen, un regrabado (refreshing) periódico, que consiste en una lectura seguida de una escritura. Las diferentes combinaciones de niveles y de flancos de las señales de control $\overline{RAS}$, $\overline{CAS}$ y $\overline{W}$ dan lugar a distintos modos de operación. Por ejemplo, en algunas memorias se realiza el regrabado manteniendo $\overline{W}$ en nivel alto, $\overline{CAS}$ en nivel bajo y aplicando flancos sucesivos de bajada a $\overline{RAS}$.

Dado que en numerosas aplicaciones, una memoria dinámica puede permanecer un tiempo elevado sin que se escriban o lean algunas posiciones, es necesario que los ciclos de regrabado se realicen de forma periódica. Por ello resulta imprescindible combinar el conjunto de células de memoria con un circuito de regrabado que suele recibir el nombre de «Controlador de Memorias Dinámicas» (Dynamic Memory Controller) (figura 7.19). En la figura 7.20 se representa el diagrama de bloques básico de un controlador de memorias dinámicas que consiste en un contador y un registro conectados con un multiplexor cuyas salidas constituyen las variables de dirección de la memoria.

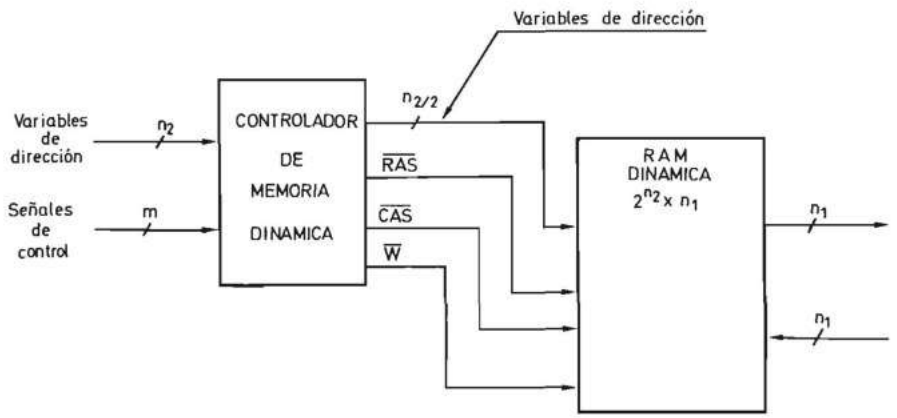


FIGURA 7.19.—**Memoria activa dinámica con circuito controlador de regrabado (refreshing).**

El contador se pone a cero con la señal adecuada (CT = 0) y cuenta los flancos de bajada sucesivos de la señal $\overline{RAS}$ (Row address strobe). El registro almacena las direcciones de las posiciones de la memoria en las que se desea realizar una lectura o escritura.

Pero la ejecución periódica y automática de ciclos de regrabado necesita un circuito de generación de señales de control que suele recibir el nombre de «Controlador de tiempos de la Memoria» (Memory Timing Controller). Dicho circuito es un sistema secuencial síncrono que genera señales de control de forma secuencial pero que admite la petición de realización de ciclos de lectura o escritura desde el exterior. Por ello su funcionamiento es similar al de la unidad de control de un procesador digital secuencial asíncrono (ver apartado 6.3.5.6.2).

La elevación de la capacidad de integración ha hecho que se hayan colocado

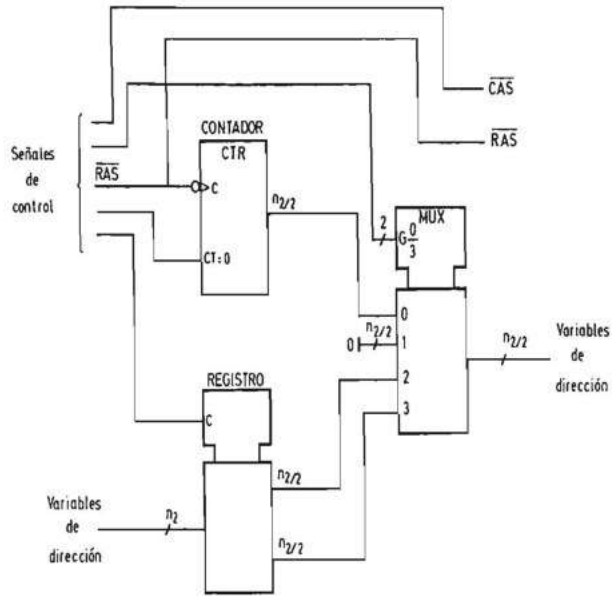


FIGURA 7.20.—**Diagrama de bloques básico de un controlador de memorias dinámicas sin generador de señales de control (Timing controller).**

en un único circuito integrado el generador de señales de control y el propio controlador de memorias dinámicas de la figura 7.20. Es usual que los fabricantes le den al conjunto el mismo nombre de controlador de memorias dinámicas.

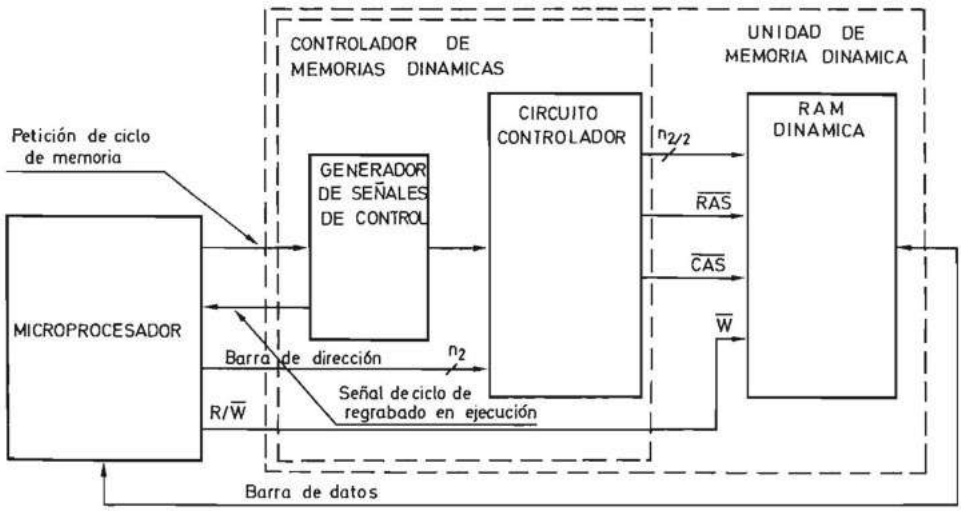


FIGURA 7.21.—**Unidad de memoria dinámica y su acoplamiento a un microprocesador.**

La principal aplicación de las memorias dinámicas es formar parte de la unidad de memoria de un procesador digital y, en particular, de los procesadores de dos niveles integrados cuya unidad central de proceso recibe el nombre de microprocesador (ver tomo II de este libro). Las infinitas formas que existen de realizar un microprocesador y, en especial, las señales que genera para relacionarse con la memoria externa (estática o dinámica) hace que el controlador de memorias dinámicas se pueda realizar de diferentes maneras.

En la figura 7.21 se representa una forma típica. El microprocesador observa la señal de «Ciclo de regrabado en ejecución» y genera impulsos de petición de ciclo de memoria cuando no se está produciendo dicha acción. Mientras no recibe petición de ciclos de memoria por parte del microprocesador, el controlador genera sucesivos ciclos de regrabado de tal manera que en el tiempo máximo especificado por el fabricante de los circuitos de memoria (típicamente algunos milisegundos), se haya producido el regrabado de todas las posiciones.

Para un estudio más profundo de las memorias dinámicas y de sus controladores así como del acoplamiento con diferentes microprocesadores se remite al lector a la bibliografía al final del capítulo. [TEXA 89] [INTE 86*a*] [NEC 87].

7.2.3.1.3.1b Memorias activas de escritura y lectura simultáneas (Read while write memories). Este tipo de memorias se representa mediante el símbolo lógico de la figura 7.22. En lugar de poseer un solo grupo de entradas de dirección para seleccionar una posición, tiene dos: uno permite seleccionar la posición que se desea leer y el otro la posición que se desea escribir. De esta forma se logra que

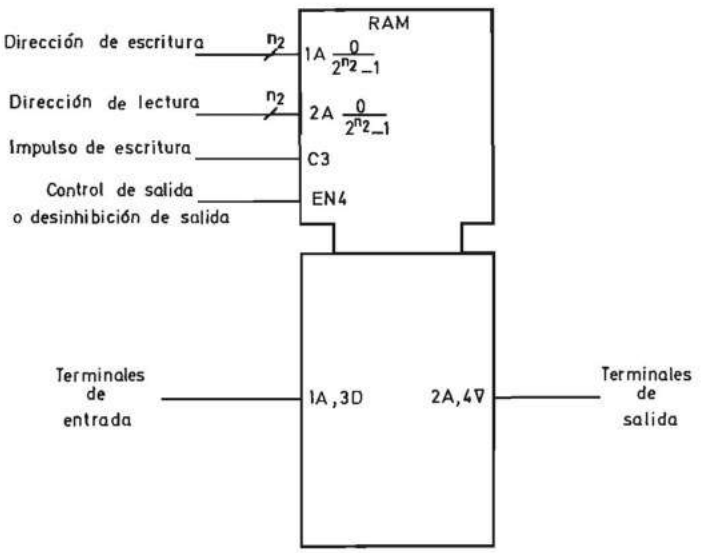


FIGURA 7.22.—**Símbolo lógico de una memoria de acceso aleatorio activa de escritura y lectura simultáneas.**

ambas operaciones se puedan realizar por medio de sistemas independientes y simultáneamente.

En la figura 7.23 se representa el diagrama de bloques de una memoria de este tipo organizada 2D. El decodificador DEC1 recibe la dirección de escritura y posee un terminal de desinhibición de las salidas al que se aplica el impulso de escritura de la memoria.

El decodificador DEC2 recibe la dirección de lectura y cada una de sus salidas se conecta a la entrada de desinhibición (EN) de n_1 puertas de tres estados que se conectan a las n_1 salidas de cada posición.

Cuando la entrada G del decodificador DEC2 se encuentra en nivel cero, las salidas de todas las puertas seguidoras se encuentran en tercer estado y, por lo tanto, la salida de la memoria se pone en el citado estado.

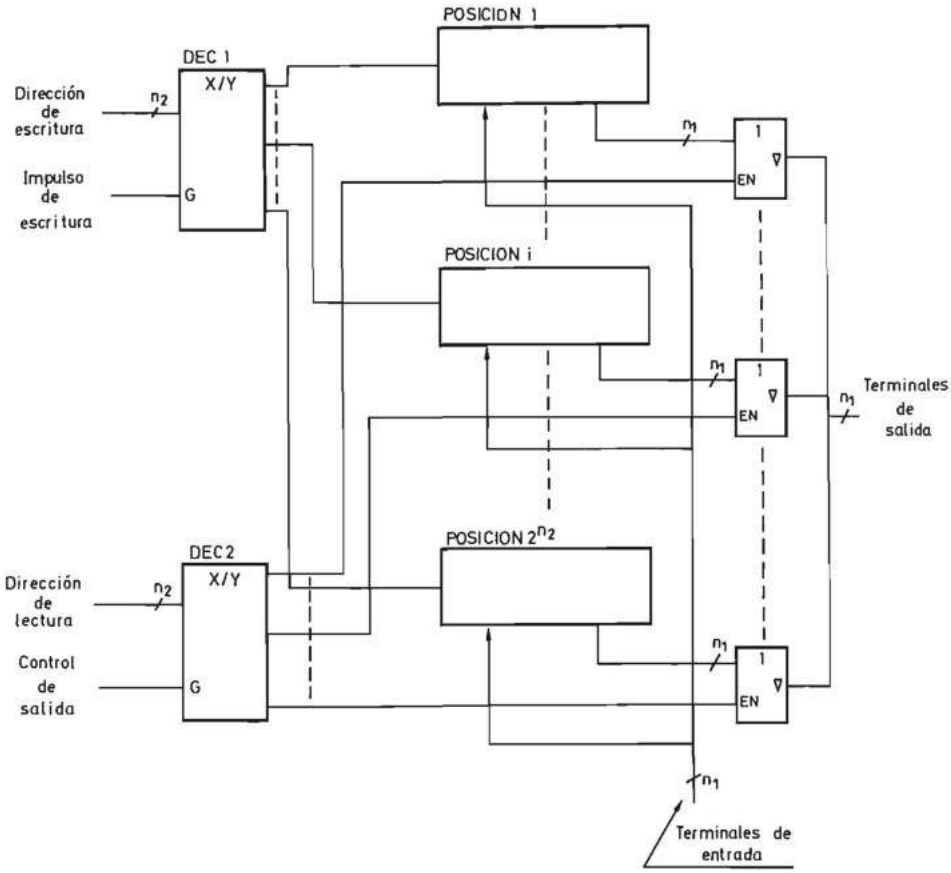


FIGURA 7.23.—**Diagrama de bloques de una memoria de acceso aleatorio activa de escritura y lectura simultáneas con organización 2D.**

FIGURA 7.24.—**Aplicación como unidad de acoplamiento de las memorias de acceso aleatorio activas de escritura y lectura simultáneas.**

Aunque estas memorias pueden ser estáticas o dinámicas, se suelen utilizar principalmente las primeras debido a la complejidad de operación que introduce la necesidad del regrabado de las segundas. No obstante, las segundas se utilizan en algunas aplicaciones en las que se tiene que producir una lectura periódica a elevada velocidad como en los controladores de tubos de rayos catódicos [RIO 90].

La aplicación más interesante de las memorias de acceso aleatorio de escritura y lectura simultáneas es la de formar parte de la unidad de acoplamiento (interfaz) entre un procesador digital y un periférico que necesita leer periódicamente la información generada por aquél. En la figura 7.24 se representa el diagrama de bloques de esta aplicación.

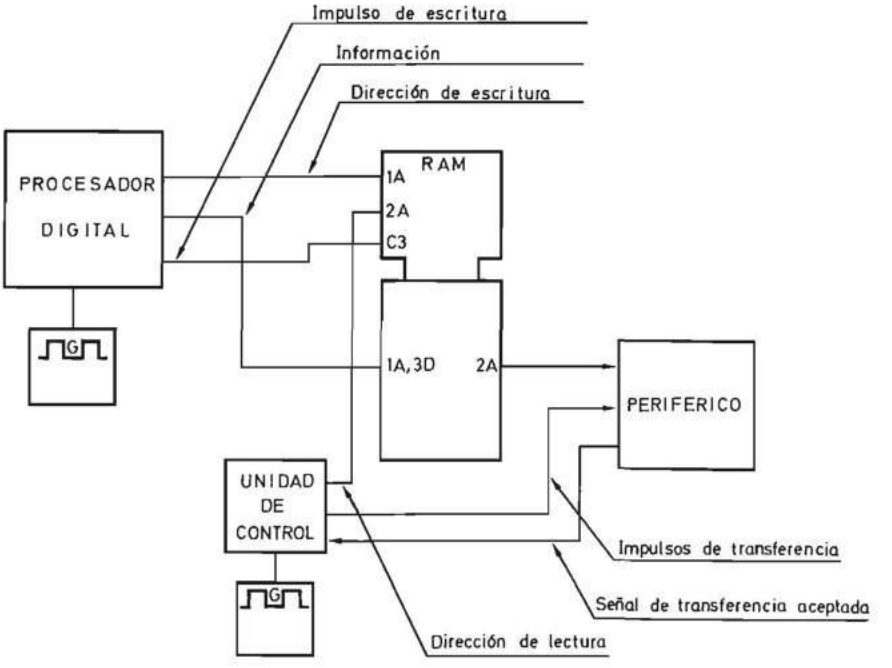


FIGURA 7.25.—**Esquema del circuito de acoplamiento de un procesador digital y un periférico, realizado con una memoria de acceso aleatorio activa de escritura y lectura simultáneas.**

En la figura 7.25 se representa un esquema más detallado. El procesador digital realiza el control de la escritura en memoria. Para ello, coloca la información en los terminales correspondientes, pone en los terminales de dirección 1A la combinación binaria que selecciona la posición de la memoria en la que quiere introducir la información, y aplica un impulso de transferencia a la entrada C3.

Por el contrario la lectura la realiza una unidad de control que genera las direcciones de lectura y aplica los impulsos de transferencia al periférico. En el caso más general el periférico informa a la unidad de control de que ha finalizado el proceso que ha de realizar con la información recibida para que aquélla pueda enviarle un nuevo dato.

En la figura 7.26 se representa una aplicación típica que es la conversión de un visualizador dinámico en estático. El procesador coloca en la memoria de acceso aleatorio la información a visualizar, consistente en caracteres alfanuméricos en el código ASCII, en el instante que quiere. El visualizador recibe a su entrada la información procedente de la memoria, cuya lectura se realiza mediante una unidad de control constituida por un contador y un decodificador. El generador de impulsos debe poseer una frecuencia adecuada para que un observador humano no detecte el parpadeo del visualizador. En este caso no es necesario ningún impulso de transferencia aceptada desde el periférico hacia la unidad de control.

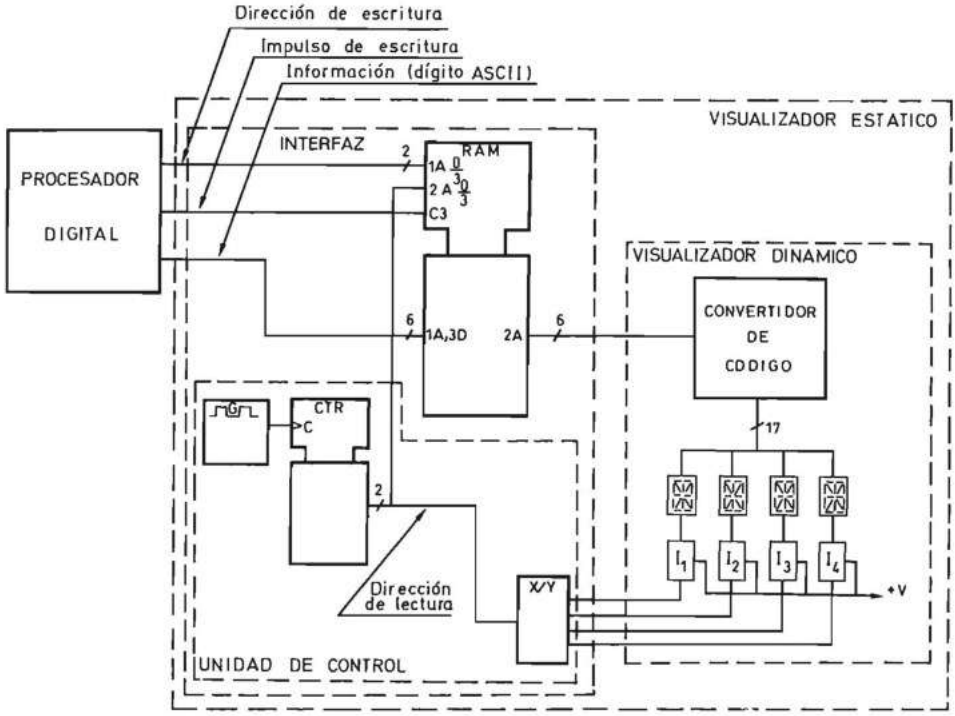


FIGURA 7.26.—**Circuito de conversión de un visualizador dinámico en estático mediante una memoria de acceso aleatorio activa de escritura y lectura simultáneas.**

Como resumen de todo lo expuesto se puede decir que las memorias de acceso aleatorio de escritura y lectura simultáneas simplifican el proceso de transferencia de información de un procesador a un periférico de salida más lento o que necesita una transferencia periódica como, por ejemplo, un visualizador dinámico o un visualizador con un tubo de rayos catódicos. Mediante una memoria de este tipo se sincronizan los sistemas digitales síncronos sin necesidad de que existan señales de control del periférico hacia el procesador, ni de que éste esté pendiente de aquél.

La unidad de acoplamiento resulta ser un procesador digital especializado y por ello suele recibir el nombre genérico de unidad de acoplamiento inteligente.

7.2.3.1.3.1c Memorias activas de acceso múltiple (Multiport memories or simultaneous access memories) Estas memorias se caracterizan porque permiten la realización de acciones de lectura, escritura o ambas en más de una posición simultáneamente. Pueden ser de acceso múltiple en la lectura, en la escritura o en ambas, y las más usuales son las de acceso doble.

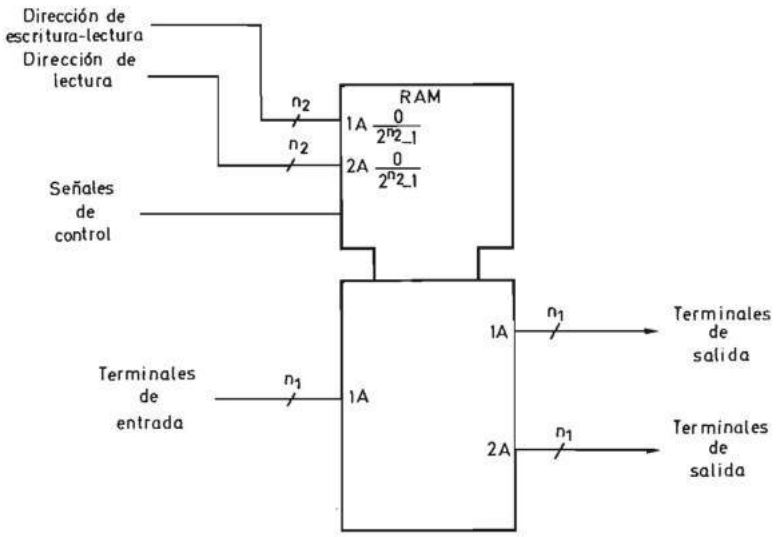


FIGURA 7.27.—**Símbolo lógico normalizado de una memoria de acceso aleatorio activada de acceso doble en la lectura.**

Las memorias de acceso doble en la lectura se obtienen simplemente duplicando los seguidores de dos entradas de la figura 7.23. Las señales de control pueden presentar diversas alternativas. Para reducir su complejidad, estas memorias suelen contener solamente dos decodificadores. Uno corresponde a uno de los canales de lectura y el otro es común al canal de escritura y al otro canal de lectura.

En la figura 7.27 se representa el símbolo lógico normalizado correspondiente a una memoria que posee:

— n_2 terminales de dirección de escritura y lectura.
— n_2 terminales de dirección de lectura.
— un conjunto de señales de control.
— n_1 terminales de entrada de información.
— n_1 terminales de salida (canal 1A).
— n_1 terminales de salida (canal 2A).

Una de las principales aplicaciones de las memorias de acceso doble en lectura es la de formar parte de las unidades operativas de un procesador digital que permite realizar operaciones con dos operandos y almacenar el resultado en la posición en la que se encuentra uno de ellos en un único ciclo del generador de impulsos. Esta aplicación se trata en profundidad en el tomo II dedicado a los procesadores digitales y a él se remite al lector.

Las memorias de acceso doble en escritura, o en escritura y lectura simultáneamente presentan una complejidad adicional debido a que deben impedir la escritura simultánea en la misma posición.

Por ello estas memorias además de un conjunto de biestables adecuadamente controlados deben poseer un circuito digital que reciba las peticiones de escritura e impida la escritura o la lectura en una posición a través de un canal si en ese instante se está escribiendo en dicha posición a través del otro canal.

En la figura 7.28 se representa el diagrama de bloques de una memoria de acceso doble en escritura y lectura que posee, por lo tanto, dos decodificadores para poder seleccionar simultáneamente dos posiciones de la memoria. Los impulsos de escritura se obtienen a través de un circuito digital que hace de árbitro (address arbitration) entre los dos sistemas que tienen acceso a la memoria. Este circuito recibe las variables de dirección y las señales de control de ambos sistemas y en

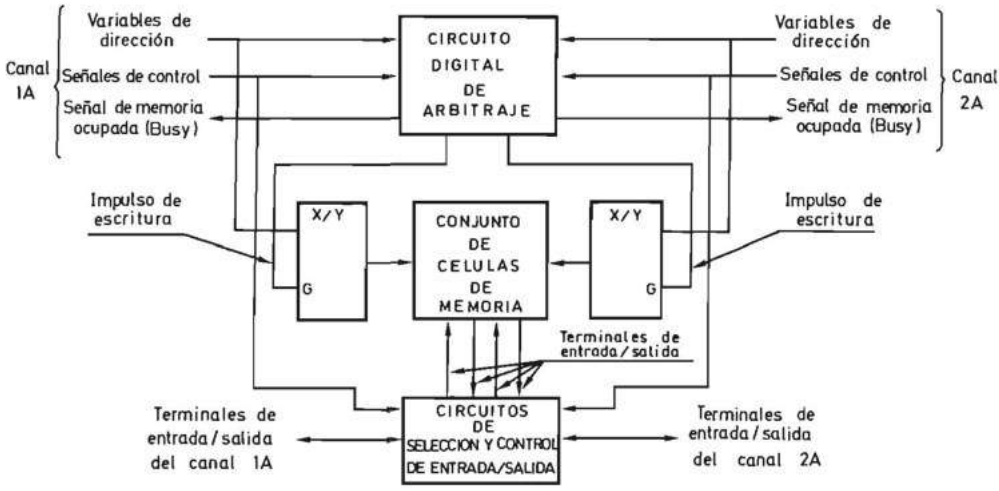


FIGURA 7.28.—**Diagrama de bloques de una memoria activa de acceso aleatorio doble en escritura y lectura.**

el caso de que ambos sistemas provoquen una operación de escritura, permite que los impulsos correspondientes lleguen a cada uno de los decodificadores sólo si ambas direcciones son diferentes. Por el contrario, si un sistema intenta leer o escribir en una posición de la memoria en la que está realizando una escritura el otro, se activa la señal de memoria ocupada (Busy) correspondiente en un intervalo mínimo para indicarle que debe esperar para realizar la transferencia. Para evitar un funcionamiento incorrecto en el caso de que ambos sistemas intenten acceder simultáneamente a la misma posición, se puede realizar el sistema físico del árbitro de tal manera que dé preferencia a uno de ellos.

La memoria posee además circuitos de entrada/salida independientes para cada canal conectados a las células de memoria.

En la figura 7.29 se representa el símbolo lógico de una memoria estática de acceso aleatorio doble en escritura y lectura. No se especifican las señales de control que recibe la memoria porque coinciden con las de una memoria de lectura y escritura no simultáneas como la de la figura 7.12 ($\overline{WE}$, $\overline{CE}$ y $\overline{OE}$). Además de ellas existe una señal de «Memoria ocupada» hacia el sistema externo.

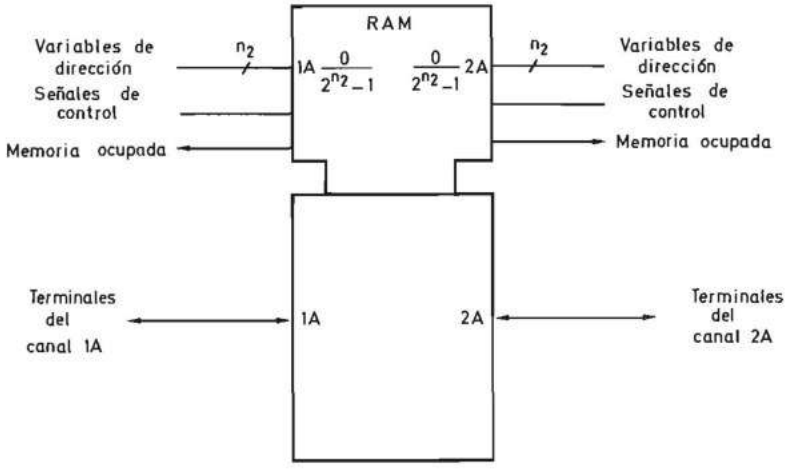


FIGURA 7.29.—**Símbolo lógico de una memoria activa de acceso aleatorio doble en escritura y lectura.**

La principal aplicación de las memorias de acceso doble en lectura y escritura es la de servir de sistema de comunicación entre dos procesadores que deban transferirse información entre ellos (figura 7.30) [INTE 86b].

7.2.3.1.3.2. Memorias pasivas Se denominan pasivas aquellas memorias en las que existe una gran diferencia entre el tiempo de lectura y el de escritura que es mucho mayor.

Una característica común a todas las memorias pasivas es la de que no son volátiles, es decir, mantienen la información aunque deje de aplicarse la tensión de

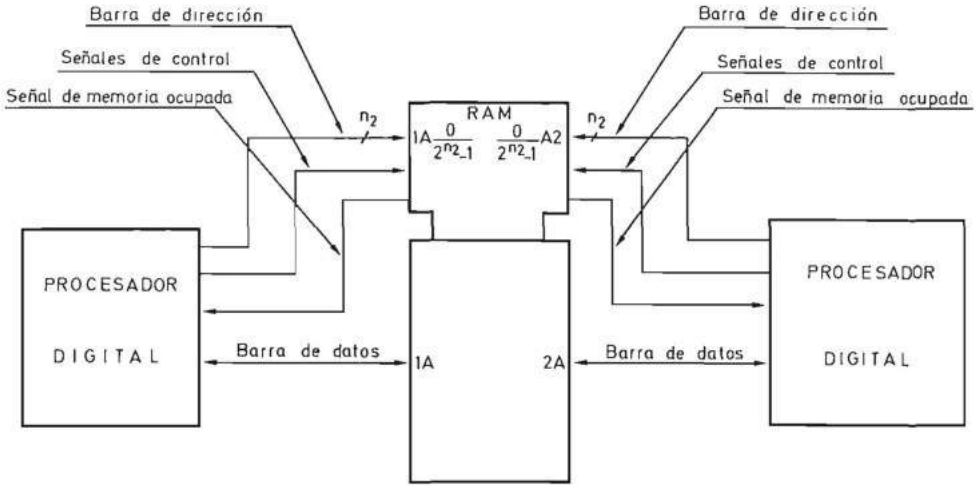


FIGURA 7.30.—**Aplicación de una memoria activa de acceso aleatorio doble en escritura y lectura, al acoplamiento de dos procesadores digitales.**

alimentación. Esta característica las hace idóneas para un conjunto de aplicaciones como son:

— Generadores de caracteres que son convertidores de código, como por ejemplo de ASCII a 5 por 7 puntos.

— Circuitos combinacionales programables, para realizar los sistemas secuenciales síncronos microprogramables.

— Memoria de instrucciones de un microcomputador o de un autómata programable.

Las memorias pasivas, se pueden clasificar en tres grandes tipos que se indican en la tabla 7.2 y se estudian a continuación.

MEMORIAS DE ACCESO ALEATORIO PASIVAS	*Totalmente pasivas*	[Read only memories (ROM)]	
	Pasivas programables	[Programable read only memories (PROM)]	
	Pasivas reprogramables [Reprogrammable read only memories (RPROM)]	— Borrables con rayos ultravioleta [Erasable programmable read only memories (EPROM)]	
		— Borrables eléctricamente	Posición a posición [Electrically erasable programmable read only memories (EEPROM)]
			Totalmente (Flash memories)

TABLA 7.2.—**Clasificación de las memorias de acceso aleatorio pasivas.**

a) Memorias totalmente pasivas [Read only memories (ROM)].

Son aquellas memorias pasivas que nunca pueden ser escritas por el usuario. Al colocarlas en un sistema digital sólo pueden ser leídas. La grabación o escritura la realiza el fabricante. En general, las células de memoria están formadas por un transistor bipolar o MOS (ver apartado 7.3.2.2) y la grabación consiste en colocarlo o no al fabricar el circuito integrado. El tiempo de lectura tiene un cierto valor que depende de la tecnología en que ha sido realizada la memoria.

En la figura 7.31 se representa el símbolo lógico típico de una memoria totalmente pasiva que posee:

— n_2 variables de dirección para seleccionar la posición de memoria cuya información aparece en los terminales de salida.

— un terminal de inhibición/desinhibición (enable) de salida que cuando se encuentra en un cierto estado (el cero en la figura 7.31), pone en tercer estado la salida de la memoria.

— n_1 terminales de salida que poseen tercer estado.

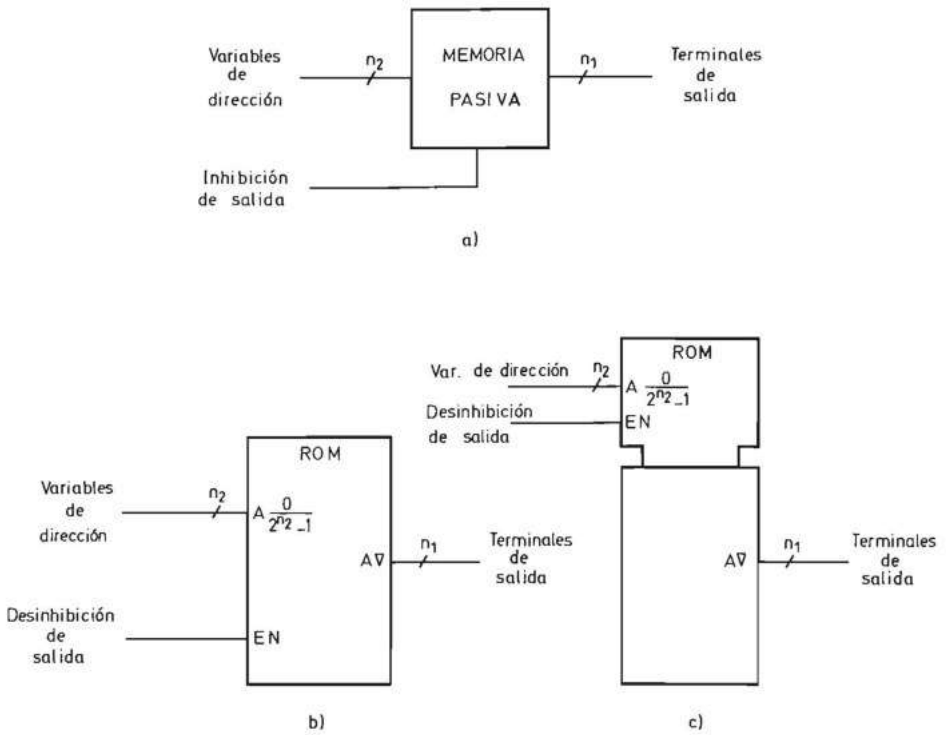


FIGURA 7.31.—**Símbolo lógico típico de una memoria de acceso aleatorio totalmente pasiva: *a*) No normalizado; *b*) y *c*) Normalizados.**

b) Memorias pasivas programables [Programmable read only memories (PROM)].

Son aquellas cuyo contenido puede ser escrito por el usuario una sola vez. En las de uso más extendido, realizadas en tecnología TTL, la célula de memoria está constituida por un diodo y la programación consiste en hacer pasar por él una corriente excesiva que lo funde (ver apartado 7.3.2.2). El tiempo de grabación es mucho mayor que el de lectura y dicha operación se realiza en un instrumento especial, que recibe el nombre de grabador y, en general, constituye un periférico de un microcomputador desde el que recibe la información a introducir en la memoria. Esta última se coloca en un zócalo de fuerza de inserción nula y una vez grabada se trasplanta al equipo electrónico.

La grabación se realiza a través de los mismos terminales de salida. Para grabar un uno en un determinado bit de una posición, se selecciona ésta, se eleva la tensión de alimentación al valor especificado por el fabricante (superior a los 5 V del TTL) y se aplica una tensión, también especificada por el fabricante al terminal de salida correspondiente.

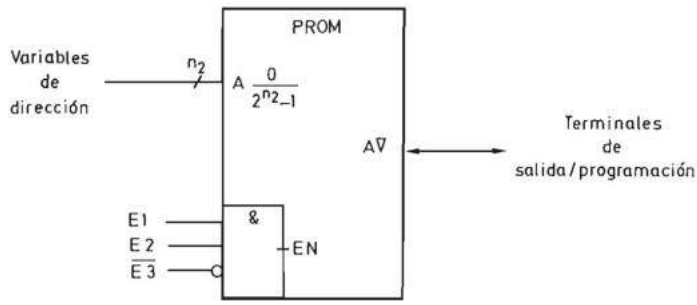


FIGURA 7.32.—**Símbolo lógico de una memoria de acceso aleatorio pasiva programable.**

En la figura 7.32 se representa el símbolo lógico de un módulo típico de memoria pasiva programable (PROM) que posee tercer estado a la salida y una variable de desinhibición EN que es el producto lógico de tres variables de entrada E1, E2 y $\overline{E3}$.

c) Memorias pasivas reprogramables. [Reprogrammable read only memories (RPROM)].

Son aquellas memorias cuyo contenido puede ser escrito por el usuario varias veces. Estas memorias han tenido una gran evolución y en la actualidad se dividen en tres grandes clases:

C-1 Memorias borrables con rayos ultravioleta [Erasable programmable read only memories (EPROM)].

Estas memorias se graban mediante impulsos eléctricos cuyo nivel de tensión es, en general, mayor que el de los normales de operación del circuito. Utilizan en su realización transistores MOS de puerta flotante que se describen en el apartado 7.3.2.2. La grabación consiste en inyectar portadores de carga en dicha puerta mediante impulsos eléctricos. El borrado se realiza mediante rayos ultravioleta y ha de preceder a la grabación; para ello el circuito se coloca bajo una zona transparente que permite el paso de los citados rayos.

Los impulsos de grabación se deben aplicar en un determinado terminal de la memoria. Las memorias EPROM han sido realizadas de varias formas de las que las principales son las siguientes:

— Memoria EPROM en las que la tensión de grabación se aplica a través de un terminal independiente (V_{pp}). A su vez existen dos versiones de este tipo:

— Memorias en las que el impulso de grabación se da a través de uno de los terminales de control de la lectura, en general, el de desinhibición [Chip enable (CE)]. El símbolo lógico de una memoria EPROM de este tipo típica se representa en la figura 7.33. En este tipo de memoria EPROM, al igual que las que se estudian a continuación (figuras 7.34 y 7.35) el terminal CE tiene dos misiones diferentes: a) Actuar sobre el tercer estado de la salida junto con OE; b) Inhibir la escritura en la memoria y reducir la potencia consumida por el circuito integrado [Power down (PWR DWN)]. Por ello este terminal tiene dos conexiones al símbolo.

— Memorias en las que el impulso de grabación se da a través de un terminal dedicado exclusivamente a dicha acción (PGM). En la figura 7.34 se representa el símbolo de una memoria EPROM de este tipo.

Para grabar una información en una posición de la memoria hay que realizar las siguientes acciones simultáneas:

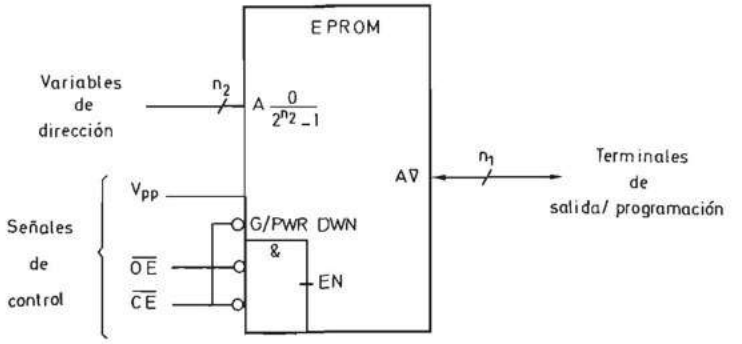


FIGURA 7.33.— **Símbolo lógico de una memoria de acceso aleatorio pasiva reprogramable EPROM con terminal independiente para la tensión de grabación (V_{pp}) e impulso de grabación a través del terminal de desinhibición ($\overline{CE}$).**

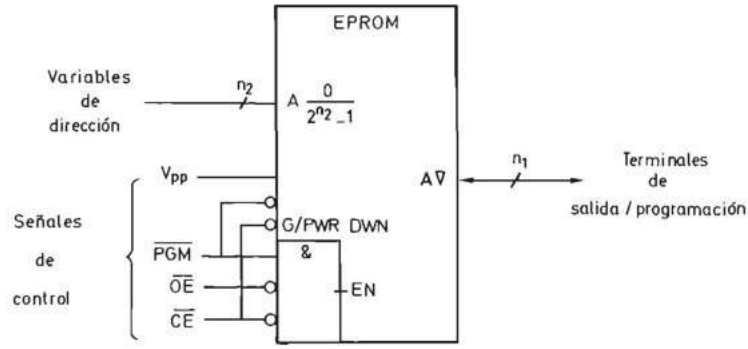


FIGURA 7.34.—**Símbolo lógico de una memoria de acceso aleatorio pasiva reprogramable EPROM con terminal independiente para la tensión de grabación (V_{pp}) y para el impulso de grabación ($\overline{PGM}$).**

— Poner en los terminales de dirección la correspondiente a la posición en la que se quiere introducir la información.
— Colocar en los terminales de salida/grabación la información a grabar.
— Poner el terminal V_{pp} en el nivel de grabación (que según la memoria varía de 13 a 25 V) (En algunas memorias es necesario además elevar la tensión de alimentación por encima del valor que hay que aplicar para leerla).
— Poner la señal $\overline{OE}$ en nivel uno.
— Poner la señal $\overline{CE}$ en nivel cero.
— Aplicar un impulso de una cierta duración al terminal $\overline{PGM}$ (la duración del impulso depende del algoritmo de grabación utilizado que se estudia más adelante).

La lectura se realiza poniendo V_{pp} a nivel de 5 V, $\overline{OE}$ y $\overline{CE}$ en nivel cero, y $\overline{PGM}$ a nivel uno.

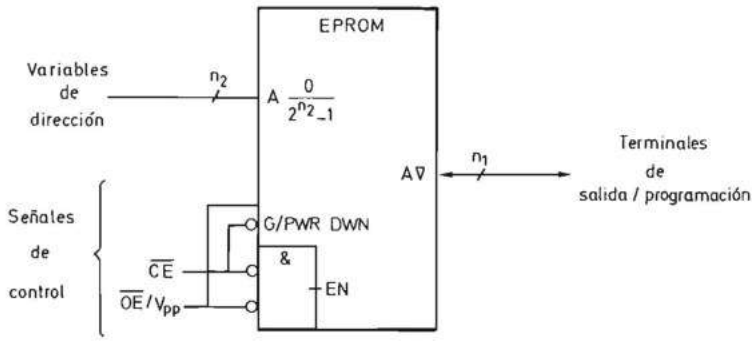


FIGURA 7.35.—**Símbolo lógico de una memoria de acceso aleatorio pasiva reprogramable EPROM con terminal único de desinhibición de salida y tensión de grabación ($\overline{OE}/V_{pp}$).**

— Memorias EPROM en las que la tensión de grabación se aplica a través de uno de los terminales de control (típicamente el de desinhibición de salida (output enable) ($\overline{OE}$ en la figura 7.35).

El fabricante de una memoria EPROM define en general mediante una tabla de funcionamiento los niveles que deben tener los distintos terminales para realizar las diferentes operaciones posibles. La tabla depende del tipo de memoria, pero resulta interesante analizar una de ellas típica como la de la tabla 7.3 que corresponde a una memoria como la de la figura 7.35.

En la tabla 7.3 se supone:

— El nivel cero corresponde a 0 V.
— El nivel uno corresponde a 5 V.
— La tensión de alimentación V_{cc} adopta un nivel V_{prog} (superior a 5 V) durante la programación y la verificación.

	Señales de control		Variables de dirección		Tensión de alimentación	Terminales salida/programación
Modo de operación	$\overline{CE}$	$\overline{OE}/V_{pp}$	A_k	A_i	V_{CC}	$D_{n_1-1} - D_0$
Lectura	0	0	X	X	5 V	Información memoria
Inhibición de salida	0	1	X	X	5 V	Tercer estado
Mínimo consumo (standby)	1	X	X	X	5 V	Tercer estado
Programación	0	V_{pp}	X	X	$V_{prog.}$	Información externa
Verificación	0	0	X	X	$V_{prog.}$	Información memoria
Inhibición de programación	1	V_{pp}	X	X	V_{prog}	Tercer estado
Identificación inteligente						
— Fabricante	0	0	V_{id}	0	5 V	Cod. Fabricante
— Dispositivo	0	0	V_{id}	1	5 V	Cod. Circuito

TABLA 7.3.—**Tabla de funcionamiento de la memoria pasiva reprogramable de la figura 7.35.**

La tabla 7.3 no presenta grandes dificultades para ser comprendida por el lector pero es conveniente describir el modo de operación «identificación inteligente» que corresponde a la última línea. Este modo de operación proporciona la posibilidad de que el sistema al que está acoplada la memoria pueda identificar su tipo,

y el fabricante. Se suele utilizar para definir este modo de operación, un determinado bit de la dirección (en la tabla 7.3 es el A_k) que se debe colocar a un nivel de tensión superior a los 5 V, que recibe la denominación de V_{id} (tensión de identificación). Un segundo bit (en la tabla 7.3 el A_i) permite distinguir entre el código del fabricante y el código del circuito.

Este modo de operación es interesante para facilitar el diseño de un grabador que identifica la memoria antes de realizar la grabación y ejecuta a continuación las acciones adecuadas al tipo identificado.

Para poder diseñar un sistema grabador de memorias EPROM es necesario conocer la relación temporal que debe existir entre las distintas señales de control, que debe ser también proporcionada por el fabricante.

En la figura 7.36 se representan los diagramas temporales de las señales de control de la memoria de la figura 7.35, correspondientes a la lectura (figura 7.36*a*) y a la grabación (figura 7.36*b*). En la figura 7.36*b* se observa que el impulso de grabación aplicado al terminal $\overline{CE}$ debe tener una determinada duración t_g, iniciarse un cierto tiempo de establecimiento t_e después de que las variables de dirección y la información externa están estables y acabar un cierto tiempo de mantenimiento t_m antes de que la tensión del terminal $\overline{OE}/V_{pp}$ pase al nivel cero. Además este último se debe producir un cierto tiempo de recuperación t_r antes de que cambie la información externa para asegurar una correcta grabación.

Para un estudio más profundo de los distintos tipos de memorias EPROM se remite al lector a la bibliografía al final del capítulo [TEXA 89], [INTE 90], [SEEQ 87].

La grabación de las memorias pasivas reprogramables EPROM debe realizarse mediante un sistema digital concebido al efecto que genere las adecuadas señales de control. Existen diversas alternativas para su realización, que se diferencian tanto en el sistema físico del grabador como en el algoritmo de grabación ejecutado.

En la figura 7.37 se representa el algoritmo más sencillo que consiste en introducir la información en las sucesivas posiciones mediante la aplicación de un impulso en la entrada de grabación (la $\overline{PGM}$ o la $\overline{CE}$ según el tipo de memoria) de una duración mínima que garantice la grabación (el valor típico es de 50 ms). Finalizada la grabación se compara el contenido de la memoria grabada con la información original para comprobar que aquélla se ha realizado correctamente.

Un grabador que ejecute este algoritmo se puede realizar mediante una memoria de acceso aleatorio activa en la que se coloca la información a grabar y una unidad de control constituida por un contador con un número de estados igual al de posiciones.

Pero muchas posiciones de una memoria EPROM necesitan un impulso de grabación de duración inferior al valor típico antes citado y, por ello, el algoritmo de la figura 7.37 resulta poco eficiente en lo que a tiempo de grabación se refiere. Por lo tanto, se puede realizar un grabador que ejecute un algoritmo más rápido que se denomina «algoritmo de grabación inteligente» y se representa en la figura 7.38. En lugar de aplicar un único impulso de gran duración a cada posición, se aplican impulsos de menor duración (valor típico de 1 ms). Después de cada impulso se compara el contenido de la memoria con la información a grabar y si no coinciden se vuelve a aplicar un impulso de 1 ms. Esta acción se repite un máximo

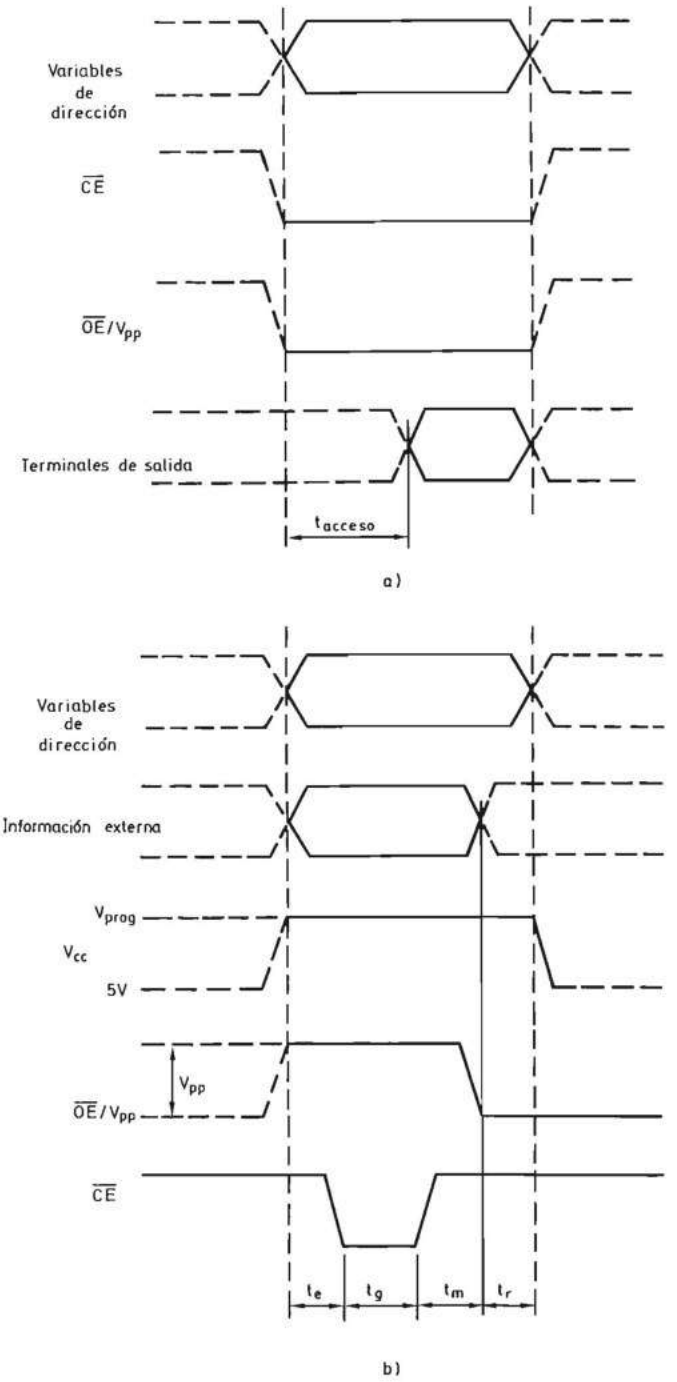


FIGURA 7.36.—**Relación temporal entre las señales aplicadas a la memoria pasiva reprogramable EPROM de la figura 7.35: *a*) Ciclo de lectura; *b*) Ciclo de grabación.**

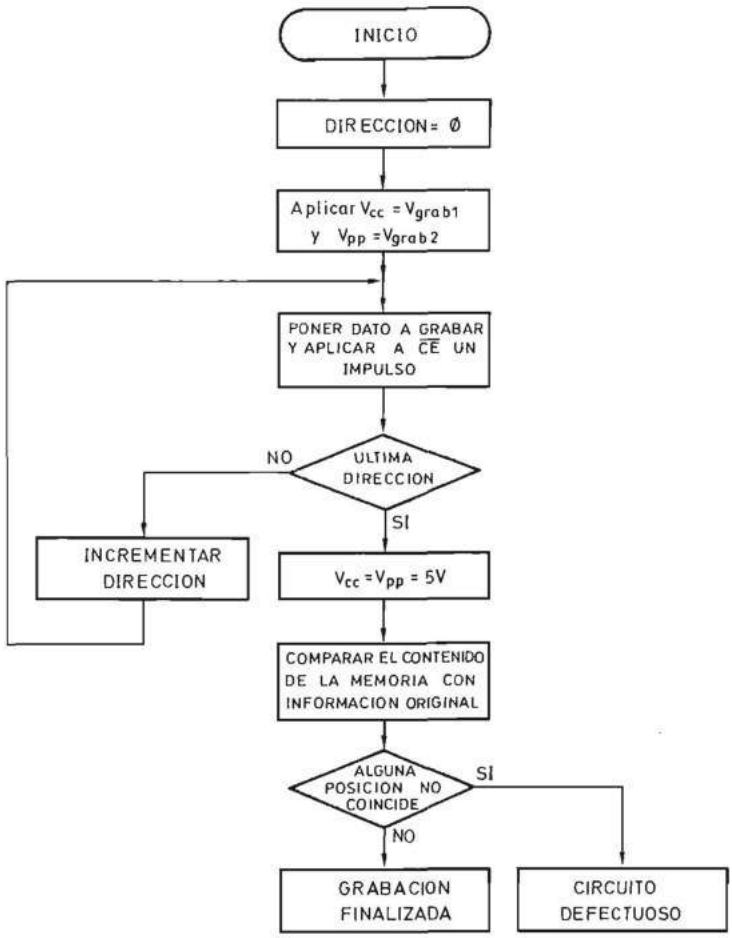


FIGURA 7.37.—**Algoritmo sencillo de grabación de una memoria pasiva reprogramable EPROM.**

de 25 veces y en el caso de que no exista coincidencia, se considera que el circuito es defectuoso. Si la coincidencia se produce antes de que se hayan aplicado 25 impulsos, se aplica un único impulso de duración igual a 3 veces el número de impulsos de 1 ms aplicados hasta ese instante para asegurar una correcta grabación. A continuación se incrementa la dirección en una unidad y se repite la secuencia de acciones. Una vez grabadas correctamente todas las posiciones se realiza una verificación del contenido de toda la memoria y si el resultado es correcto se da por finalizada la grabación.

La todavía excesiva duración del «algoritmo de grabación inteligente» junto con la disminución de la duración mínima de los impulsos de grabación propició el desarrollo de un algoritmo más eficiente que recibe el nombre de «algoritmo rá-

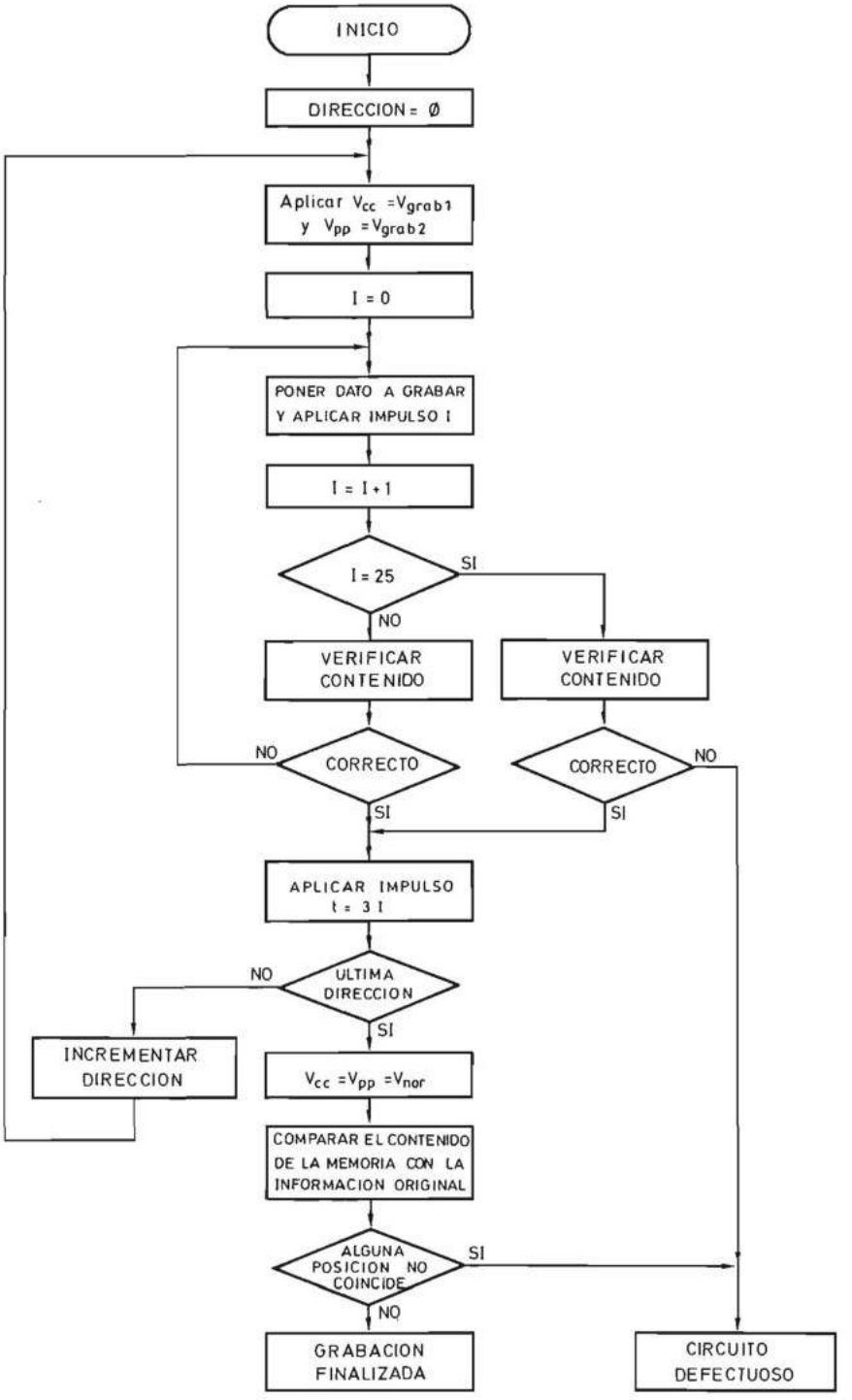


FIGURA 7.38.—**Algoritmo inteligente de grabación de una memoria pasiva reprogramable EPROM.**

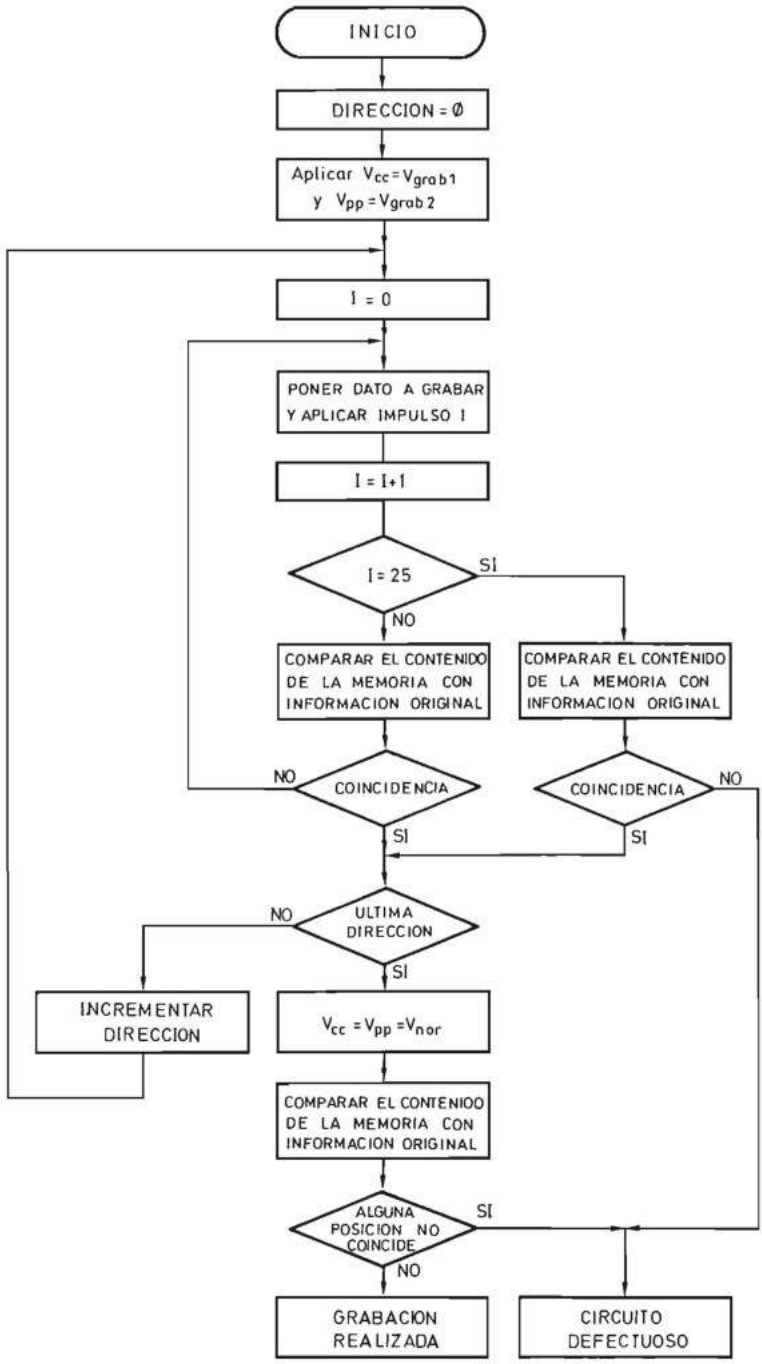


FIGURA 7.39.—**Algoritmo rápido de grabación de una memoria pasiva reprogramable EPROM.**

pido» (Quick-pulse programming algorithm) y que se representa en la figura 7.39. La duración del impulso I es menor (típicamente de 100 μs) y no es necesario aplicar un impulso adicional para garantizar la correcta grabación.

Tanto el algoritmo de grabación inteligente como el rápido no se pueden realizar con un grabador cuya unidad de control sea un simple contador, y se necesita un procesador que contenga un programa que ejecute alguno de los algoritmos de las figuras 7.38 y 7.39. La solución más utilizada es un microcomputador (figura 7.40*a*) que posee como periféricos uno o más zócalos de fuerza de inserción nula conectados por medio de la unidad de acoplamiento adecuada a través de la cual se aplican a la memoria EPROM situada en el zócalo los impulsos de duración preestablecida.

Un grabador de memorias EPROM se puede representar mediante un bloque en el que se indican el visualizador, el teclado y el zócalo para la memoria (figura 7.40*b*).

La utilización de un microcomputador para la realización del grabador aporta entre otras las siguientes ventajas:

— Permite grabar diferentes memorias que necesitan señales de grabación distintas unas de otras.

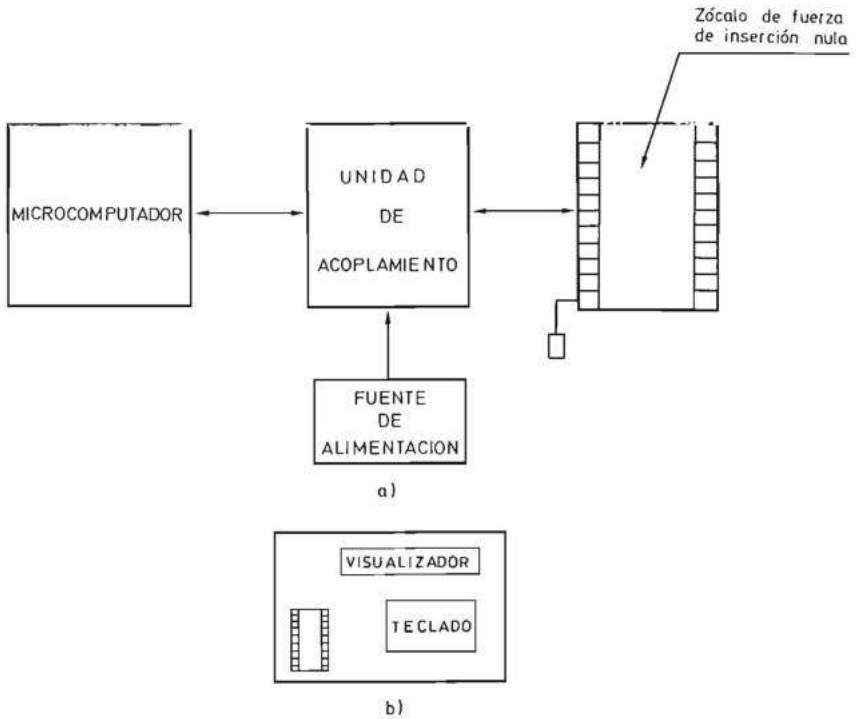


FIGURA 7.40.—**Grabador de memorias pasivas reprogramables EPROM: *a*) Diagrama de bloques; *b*) Bloque funcional.**

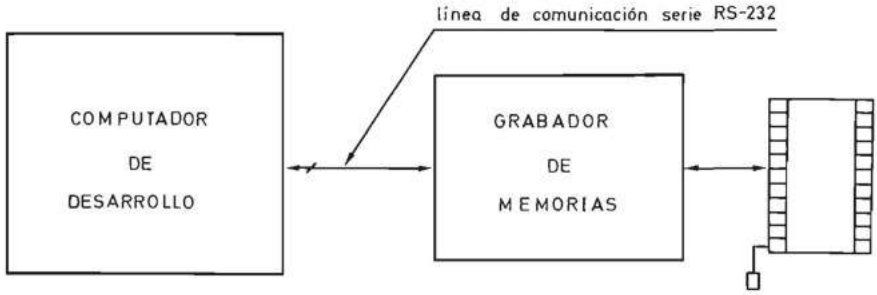


FIGURA 7.41.—**Grabador de memorias pasivas reprogramables EPROM conectado a un microcomputador de desarrollo.**

— Permite la grabación de varias memorias en paralelo (utilizando varios zócalos).

— Mediante la dotación de una unidad de comunicación permite grabar la memoria EPROM a partir de la información procedente de otro computador que, en general, es un sistema de desarrollo de programas para un determinado microprocesador (figura 7.41). En la figura 7.42 se representa un grabador comercial que posee todas las características citadas.

La principal aplicación de las memorias EPROM es la de contener las instrucciones de un microcomputador cuyo diagrama de bloques se representa en la figura 7.43.

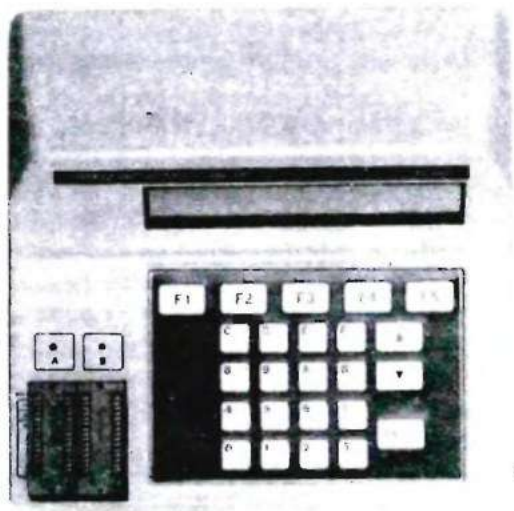


FIGURA 7.42.—**Grabador de memorias pasivas reprogramables EPROM (Cortesía de Promax, S.A.).**

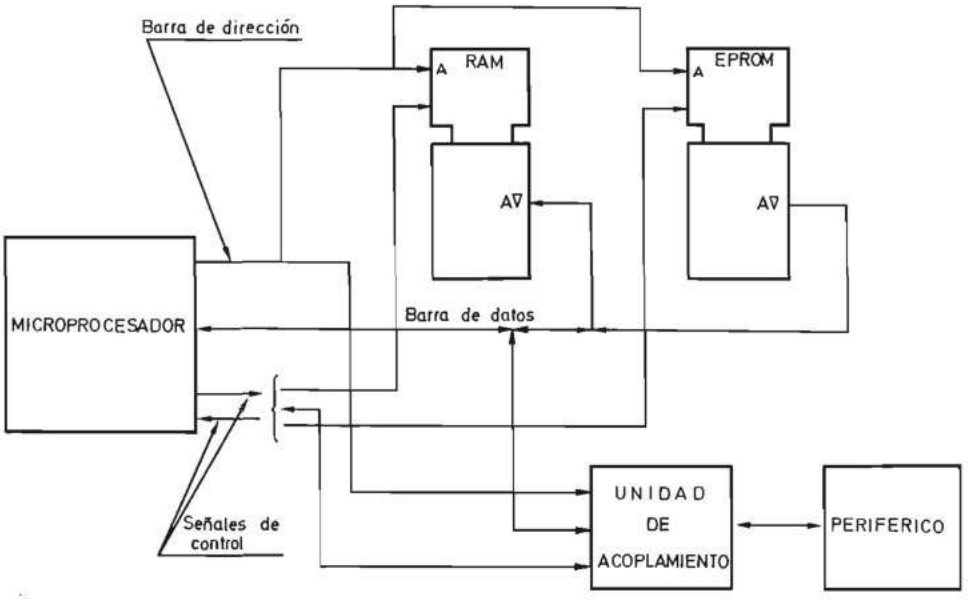


FIGURA 7.43.—**Aplicación de las memorias pasivas reprogramables EPROM al almacenamiento de las instrucciones de un microcomputador.**

C-2) Memorias pasivas reprogramables borrables eléctricamente [Electrically erasable programmable read only memories (EEPROM)].

Las memorias reprogramables borrables mediante rayos ultravioleta estudiadas en el apartado anterior, presentan la característica de que no puede modificarse la información contenida en ellas mientras están colocadas en el circuito que las utiliza.

Pero existen aplicaciones en las que dicha modificación resulta muy interesante. Por ello los fabricantes de circuitos integrados desarrollaron transistores MOS de puerta flotante que permiten su borrado mediante señales eléctricas. Esto dio lugar a las memorias borrables eléctricamente que se describen físicamente en el apartado 7.3.2.2.

Para realizar la grabación y el borrado de las memorias EEPROM es necesario aplicar a los transistores de puerta aislada impulsos de tensión superior a los que se aplican para leerlas (típicamente 20 V en el primer caso y 5 V en el segundo). Por ello las memorias EEPROM se pueden realizar de dos maneras que han dado lugar a otras tantas versiones:

— Memorias EEPROM que poseen un terminal para aplicar los impulsos de grabado (V_{pp}).

Su diagrama de bloques se representa en la figura 7.44. Además de los terminales de control $\overline{CE}$ y $\overline{OE}$ la memoria posee un terminal V_{pp}. Durante la lectura, V_{pp} se ha de alimentar a 5 V pero durante la escritura o el borrado se ha de ali-

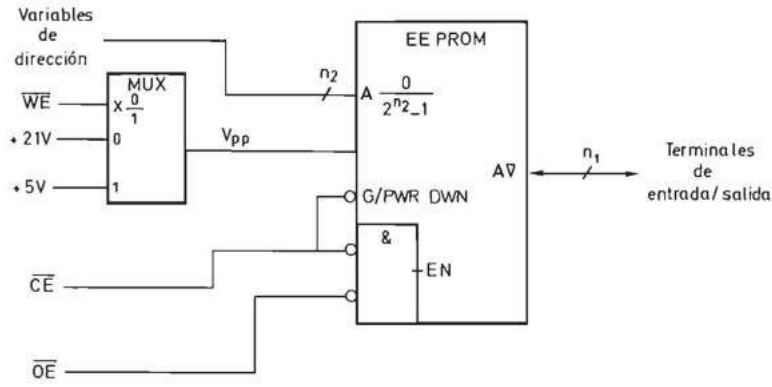


FIGURA 7.44.—**Símbolo lógico de una memoria pasiva reprogramable EEPROM con terminal de impulsos de grabación (V_{pp}).**

mentar a 21 V. Por ello es necesario colocar un circuito conmutador analógico externo (figura 7.44). Las primeras memorias EEPROM corresponden a esta versión. Al lector interesado en estudiarlas con mayor detalle se le remite a la bibliografía [INTE 81].

— Memorias EEPROM con una única tensión de alimentación y generación interna de las tensiones de alimentación. Su diagrama de bloques se representa en la figura 7.45*a*. En él se incluye además del conjunto de células, un multiplexor analógico y un convertidor de 5 V a los 21 V que necesita la memoria para su borrado o escritura. Se obtiene así el símbolo lógico de la figura 7.45*b* que corresponde a una memoria EEPROM que se alimenta a 5 V [INTE 86] [SEEQ 87].

Estas memorias suelen presentar además un modo de operación de borrado de toda la memoria, que se obtiene aplicando a la señal de control $\overline{OE}$ una tensión superior a 5 V (especificada por el fabricante) y al mismo tiempo un nivel activo a la señal $\overline{WE}$ durante el tiempo adecuado.

Las memorias EEPROM utilizan dos transistores por bit en lugar de uno solo de las EPROM (ver apartado 7.3.2.2). Por ello no se han utilizado en general para almacenar las instrucciones de un procesador sino aquellos datos (por ejemplo, tablas) que se necesita modificar en algún momento de la ejecución del proceso. Por lo tanto, la memoria EEPROM de la figura 7.45 puede constituir una parte de la memoria de un procesador y conectarse a él mediante el circuito de la figura 7.46, que utiliza un biestable *R-S* formado por las puertas NO-Y (NAND) N1 y N2 que permite al procesador poner la memoria en estado de escritura o borrado ($\overline{WE}$ = 0) o de lectura ($\overline{WE}$ = 1). Para realizar el borrado o la grabación de una posición es necesario seleccionar la posición y poner $\overline{CE}$ en estado cero. La selección de borrado o grabación se realiza mediante la señal $\overline{OE}$. Si se pone a nivel uno, la barra de datos se pone en tercer estado y se lleva a cabo el borrado; si por el contrario $\overline{OE}$ está en nivel cero, se introduce en la posición de memoria la información situada en la barra de datos.

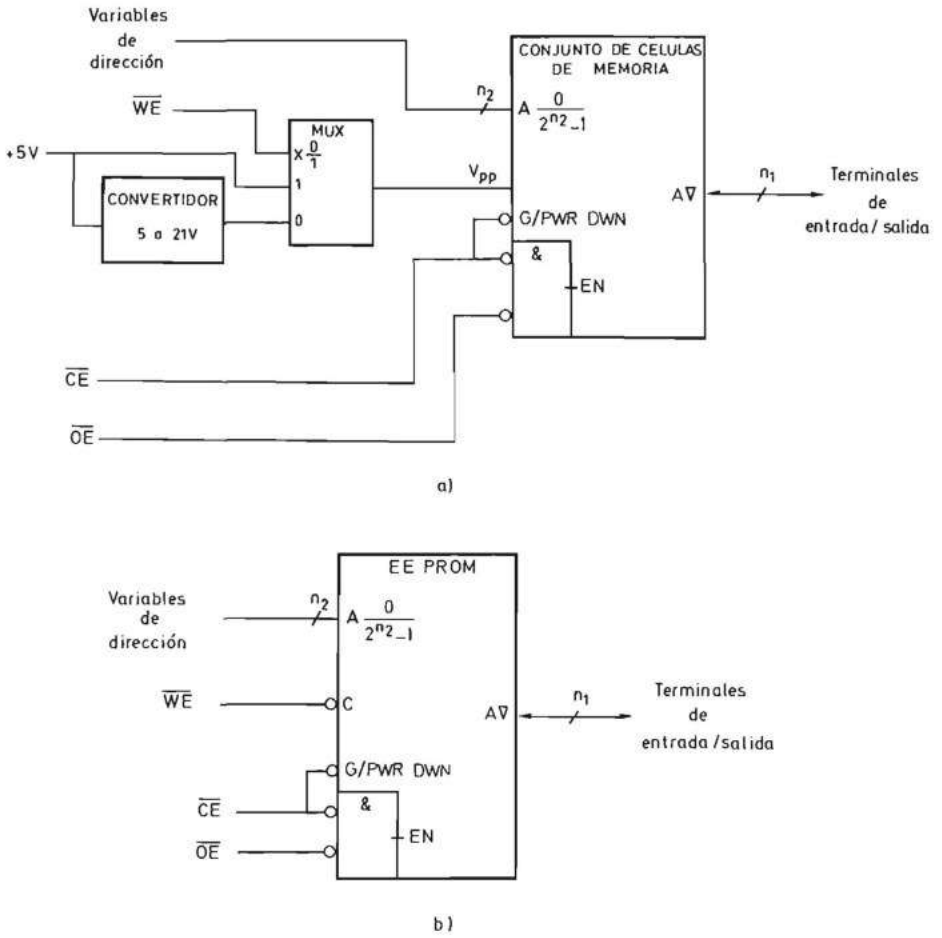


FIGURA 7.45.—**Memoria pasiva reprogramable EEPROM con tensión de alimentación única: *a*) Diagrama de bloques; *b*) Símbolo lógico.**

Para realizar la grabación o borrado es necesario que la señal $\overline{WE}$ permanezca en nivel cero un cierto tiempo (del orden de algunos milisegundos). Para ello el procesador ha de ejecutar una rutina de grabación/borrado como la de la figura 7.47.

Para evitar la necesidad de utilizar la rutina de la figura 7.47, se desarrollaron las memorias EEPROM cuyo circuito se representa en la figura 7.48*a* que coincide con la figura 7.45*a* con la única diferencia de poseer un temporizador de escritura cuya salida se activa (pasa a nivel uno) durante un cierto tiempo a partir del instante en que se aplica un nivel cero a la entrada $\overline{WE}$ [INTE 86]. Durante el intervalo de temporización se pone a cero la señal «Grabación realizada» (Ready) y su paso

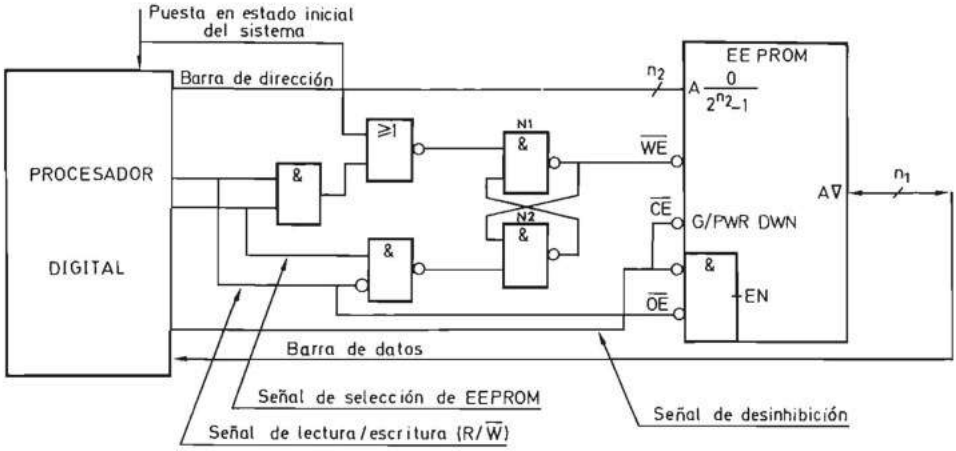


FIGURA 7.46.—**Circuito de conexión de la memoria pasiva reprogramable EEPROM de la figura 7.45 a un procesador digital.**

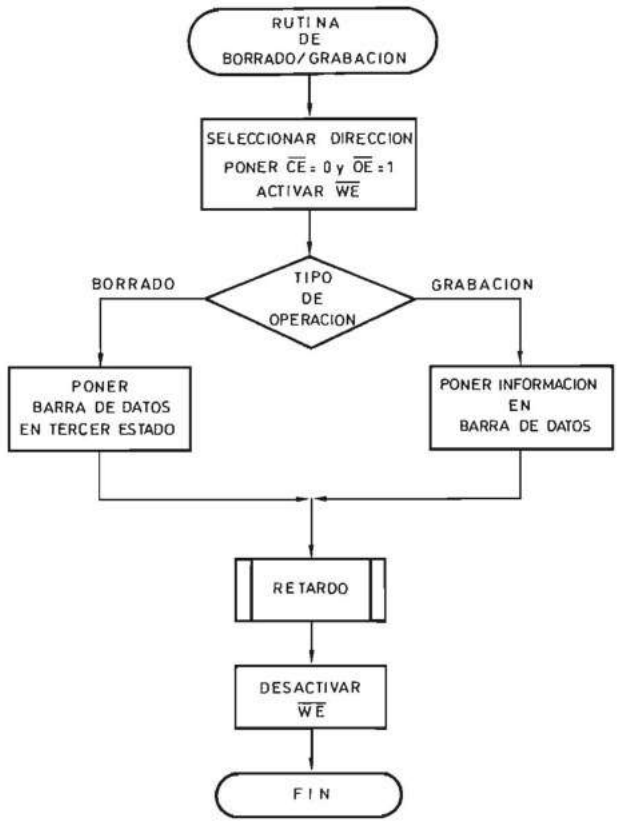


FIGURA 7.47.—**Algoritmo de grabación de la memoria de acceso aleatorio pasiva reprogramable EEPROM de la figura 7.45.**

al nivel uno indica al procesador la realización de la grabación. La señal de grabación realizada puede parar al procesador o bien interrumpirlo (ver tomo II). En la figura 7.48*b* se representa el símbolo lógico de una memoria EEPROM con temporizador de grabación incorporado y en la figura 7.49 su acoplamiento a un procesador digital.

Tal como se indicó anteriormente, la principal aplicación de las memorias EEPROM es la de formar parte, junto con las memorias activas (RAM) y pasivas EPROM, de la memoria de un microcomputador, para almacenar tablas modificables por el propio sistema.

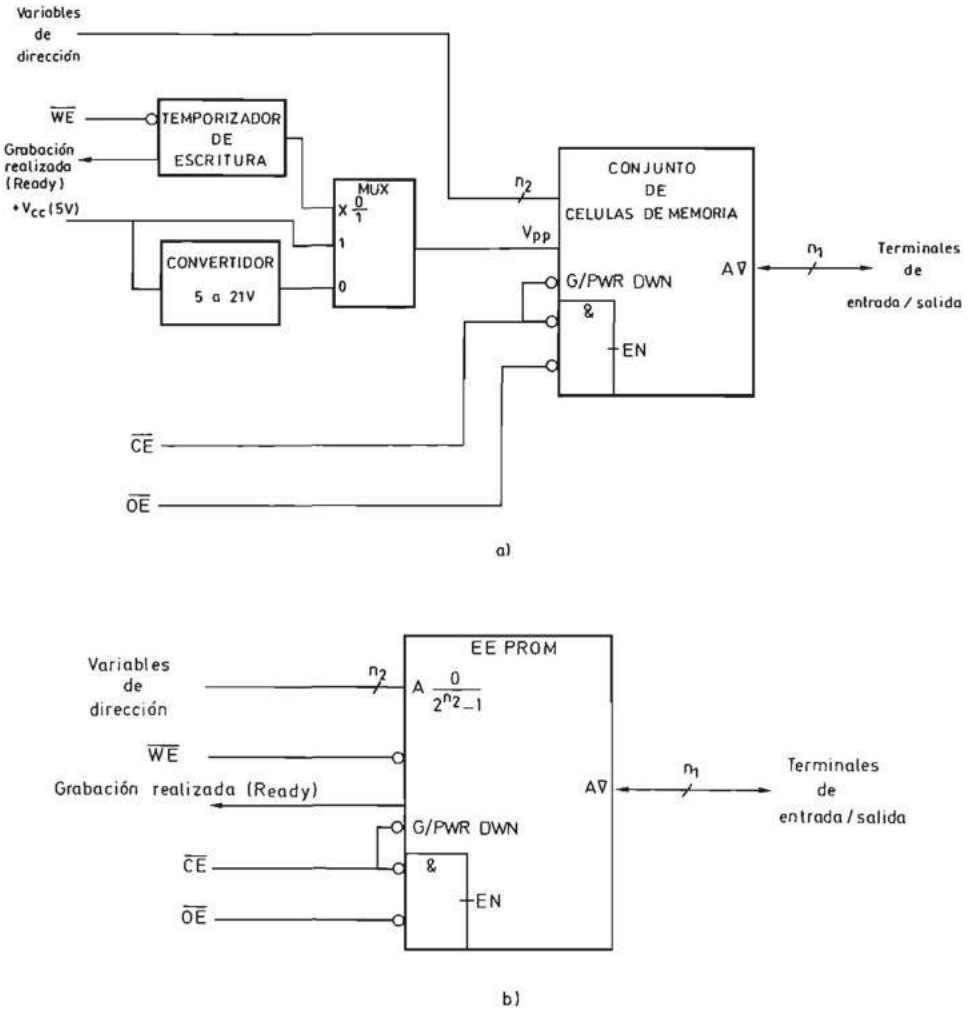


FIGURA 7.48.—**Memoria pasiva reprogramable EEPROM con temporizador de escritura incorporado: *a*) Diagrama de bloques; *b*) Símbolo lógico.**

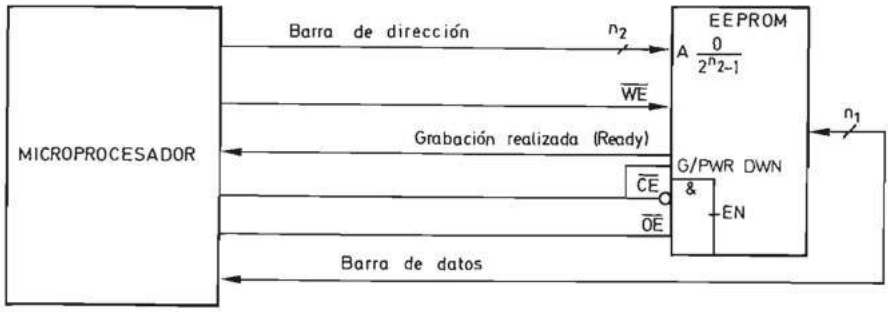


FIGURA 7.49.—**Conexión de la memoria pasiva reprogramable de la figura 7.48 a un microprocesador.**

En muchas ocasiones esas tablas se transmiten desde un computador principal y, por ello, en ese caso el microcomputador posee un diagrama de bloques como el de la figura 7.50, en el que uno de los periféricos es un MODEM, a través del cual el microprocesador recibe la nueva información a grabar en la memoria EEPROM.

En el microcomputador de la figura 7.50 la memoria pasiva reprogramable EEPROM se conecta al microprocesador a través de las mismas barras (buses) que la memoria activa (RAM) y la memoria pasiva reprogramable EPROM. Esta arquitectura es adecuada cuando el microcomputador se implementa mediante circuitos integrados independientes, cada uno de los cuales contiene el microprocesador, la memoria activa (RAM), la memoria pasiva reprogramable EPROM y la memoria

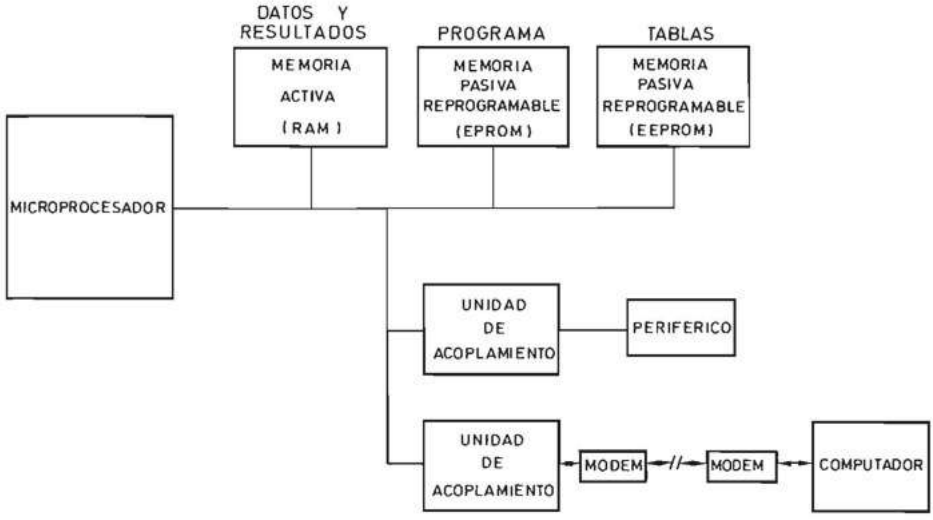


FIGURA 7.50.—**Aplicación de las memorias pasivas reprogramables EEPROM al almacenamiento de tablas.**

pasiva reprogramable EEPROM. Pero en numerosas aplicaciones, como por ejemplo la realización de sistemas electrónicos de control de muchos productos industriales (robots, televisores, equipos de comunicaciones, etc.) se utiliza un microcomputador integrado o microcontrolador, que es un circuito integrado de muy gran escala de integración que contiene en su interior el microprocesador, la memoria activa y la memoria pasiva reprogramable EPROM. En determinados casos es necesario además, utilizar una memoria pasiva reprogramable EEPROM situada en un circuito integrado externo para almacenar resultados o datos que puedan ser modificados en cualquier instante y que no desaparezcan al dejar de aplicarse tensión al circuito. Para poder dedicar el máximo número posible de terminales del circuito integrado microcontrolador a la conexión de los periféricos de entrada y salida, la solución más adecuada es aquella que minimiza el número de terminales necesarios para escribir y leer el contenido de la memoria pasiva reprogramable EEPROM. Por esta razón algunos fabricantes de circuitos integrados han desarrollado memorias de acceso aleatorio EEPROM cuya lectura y escritura se realiza a través de un canal o barra (bus) de comunicación serie (por ello reciben la denominación en inglés de «Serial EEPROM») tal como se indica en la figura 7.51. En la figura 7.52 se representa el diagrama de bloques de una memoria de este tipo, que denominamos EEPROM serie. En ella se observa que además del bloque de memoria propiamente dicho, posee un procesador de comunicaciones que se encarga de gestionar la transferencia de información entre la memoria y el microcontrolador externo. El procesador de comunicaciones recibe en serie, procedente del microcontrolador, el código de la operación que debe ejecutar así como la dirección de la memoria sobre la que debe realizar las acciones correspondientes, y además en el caso de una operación de escritura recibe también en serie la información que se debe memorizar. A continuación genera todas las señales necesarias para escribir en una posición de la EEPROM durante una operación de escritura y para leer su contenido durante una

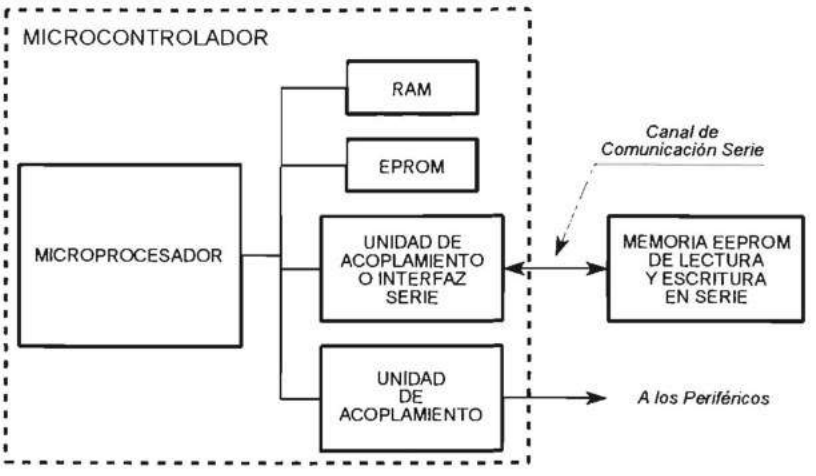


FIGURA 7.51.—**Conexión de un microcontrolador y de una memoria EEPROM de lectura y escritura en serie.**

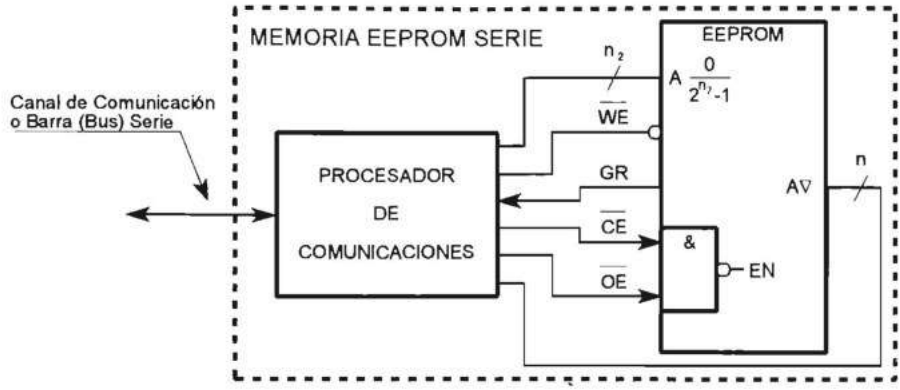
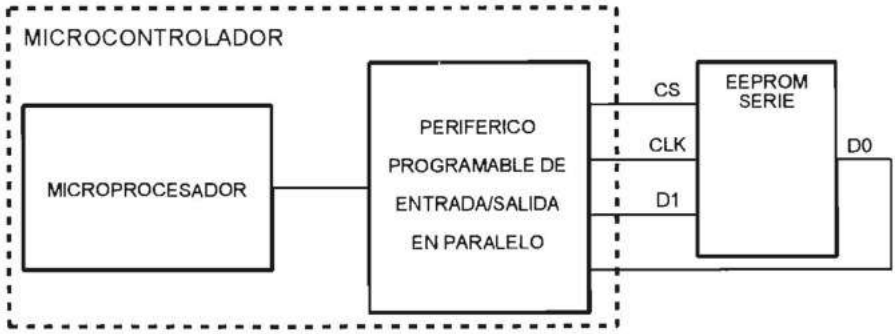


FIGURA 7.52.—**Diagrama de bloques de una memoria EEPROM de lectura y escritura en serie.**

operación de lectura. Existen muchas formas de realizar una memoria EEPROM serie que se diferencian tanto por las operaciones que se pueden ejecutar con ella como por la forma de implementar el canal de comunicación serie. En la figura 7.53 se representa una solución típica (registrada mediante la marca «Microwire» por National Semiconductor Incorporated) en la cual la memoria EEPROM serie posee una entrada de inhibición [en inglés «Chip Select» (CS)], una entrada de impulsos de transmisión serie [en inglés «Clock» (CLK)], una entrada de datos [en inglés «Data Input» (DI)] y una salida de datos [en inglés «Data Output» (DO)] que además se utiliza como señal de grabación realizada en una operación de escritura [en inglés «Ready/Busy» (RDY/BSY)]. El microcontrolador genera o lee las señales antes indicadas a través de un puerto de entrada/salida en paralelo. En la figura 7.54 se representa la relación temporal entre las diferentes señales de los terminales de



CS = Selección de circuito (Chip Select)
CLK = Impulsos de transmisión serie
D1 = Entrada de dato (Data Input)
D0 = Salida de dato/estado (Data Output)

FIGURA 7.53.—**Ejemplo de conexión de una EEPROM serie a un microcontrolador.**

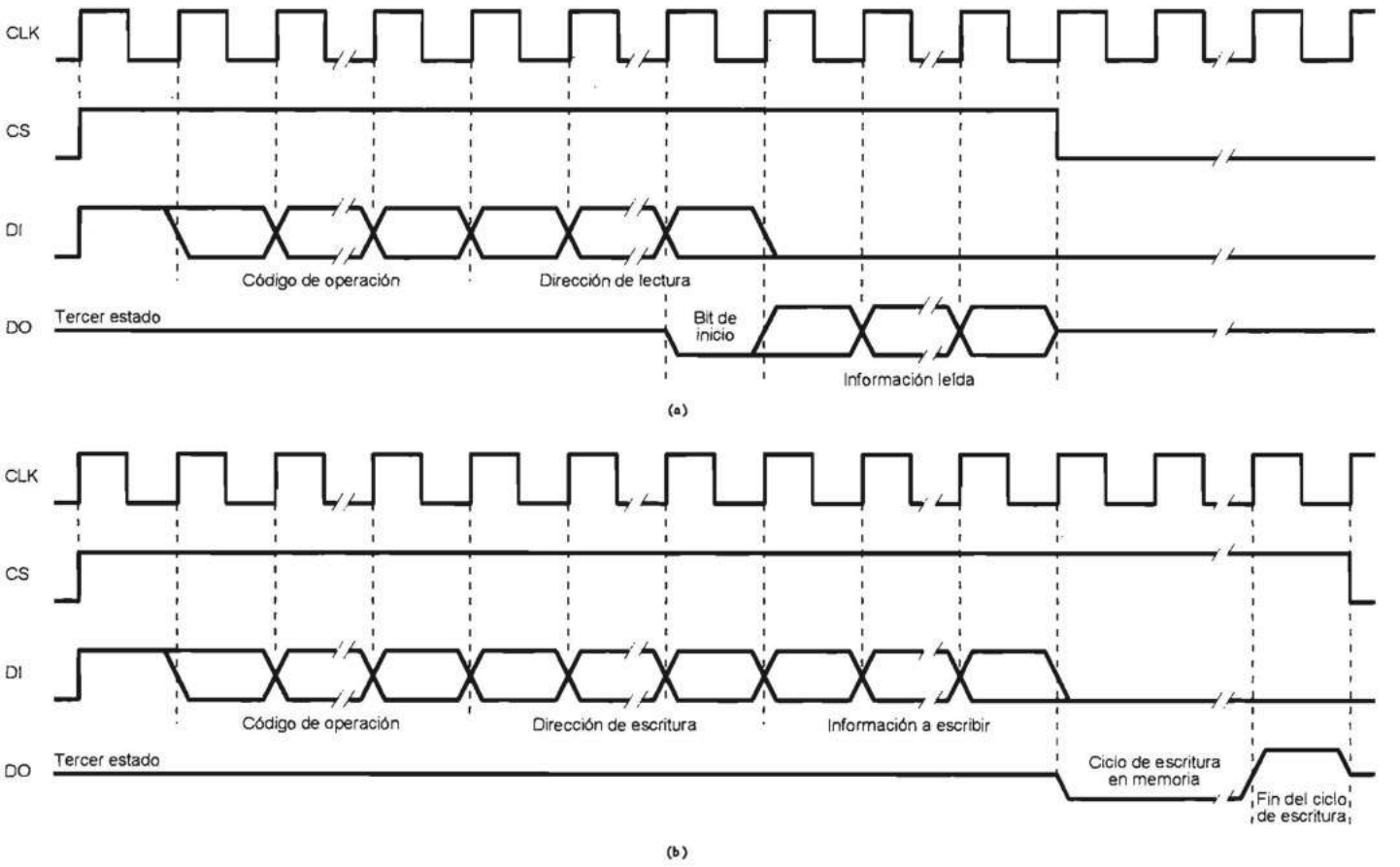


FIGURA 7.54.—**Diagrama de secuencia de las señales de una memoria EEPROM: *a*) ciclo de lectura; *b*) ciclo de escritura.**

la memoria en una operación de escritura y en una operación de lectura. Los impulsos de transmisión se aplican de forma continuada al procesador de comunicaciones de la memoria pero éste no realiza acción alguna hasta que recibe un flanco de subida en CLK al mismo tiempo que CS y DI se encuentran en nivel uno. A partir de ese instante el microcontrolador envía al procesador de comunicaciones de la memoria, a través de DI, el código de operación bit a bit en sucesivos flancos de los impulsos de transmisión, seguido de la dirección de memoria si se trata de una operación de lectura o de escritura. Durante ese tiempo el terminal DO permanece en tercer estado. A continuación, en el caso de una lectura (figura 7.54*a*), el procesador de comunicaciones coloca en el terminal de salida DO, durante sucesivos ciclos de los impulsos de transmisión, los bits de la posición de memoria seleccionada para que sean leídos por el microcontrolador. El terminal DO se pone a cero durante el ciclo que precede a la transmisión de la información leída). Por el contrario en el caso de una escritura (figura 7.54*b*) el procesador de comunicaciones lee a través de DI los bits a memorizar y, una vez recibido el último, coloca el terminal DO en estado cero mientras ejecuta la operación de escritura para indicar al microcontrolador que está ocupado (en inglés «Busy») y que no puede recibir una nueva orden de operación. Finalizada la operación de escritura, el procesador de comunicaciones pone a uno el terminal DO durante un ciclo de los impulsos de transmisión, para indicarle al microcontrolador que ya esta preparado (en inglés «Ready») para recibir una nueva orden de operación.

C-3) Memorias borrables eléctricamente en su totalidad (Flash memories).

La característica de las memorias EEPROM indicada en el apartado anterior, de poseer una densidad mucho menor que las EPROM, que impidió su utilización generalizada como memoria de programa, limitó su campo de aplicación.

Por otra parte, el progreso de la microelectrónica propició el desarrollo de sistemas de control distribuido constituidos por un conjunto de microcomputadores conectados en red que cooperan en la realización de una tarea compleja. Esto trajo consigo el interés de disponer de microcomputadores cuyo programa pueda ser modificado periódicamente, lo que no permite que la memoria utilizada sea del tipo EPROM.

Todo ello motivó el interés de los fabricantes por el desarrollo de memorias pasivas borrables eléctricamente que poseyesen la misma densidad que las EPROM. Ello dio lugar al desarrollo de las memorias tipo flash (Flash memories) [INTE 90] que son memorias borrables eléctricamente solamente en su totalidad o por bloques (en el apartado 7.3.2.2 se analiza la célula básica de las memorias FLASH).

En la figura 7.55 se representa el símbolo lógico típico de una memoria Flash que es similar al de las memorias EPROM estudiadas anteriormente y en la figura 7.56 su diagrama de bloques simplificado. En este diagrama se observa que la memoria FLASH posee, además de las células y el circuito generador de las señales de grabación y borrado, un conjunto de elementos que tienen como finalidad reducir al mínimo las señales externas de control y facilitar el acoplamiento a un microprocesador.

Los principales elementos representados en la figura 7.56 son:

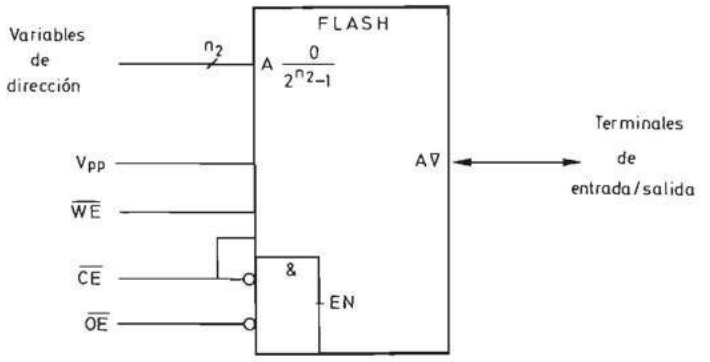


FIGURA 7.55.—**Símbolo lógico de una memoria pasiva reprogramable FLASH.**

a) Una unidad de control que a partir de los impulsos de escritura ($\overline{WE}$) y la señal de desinhibición de grabación y lectura ($\overline{CE}$) genera las señales de control necesarias para los demás elementos.
b) Un registro de direcciones que memoriza temporalmente la dirección durante los procesos de grabación y lectura.
c) Un registro de datos que memoriza temporalmente la información que se ha de grabar en una posición de la memoria.

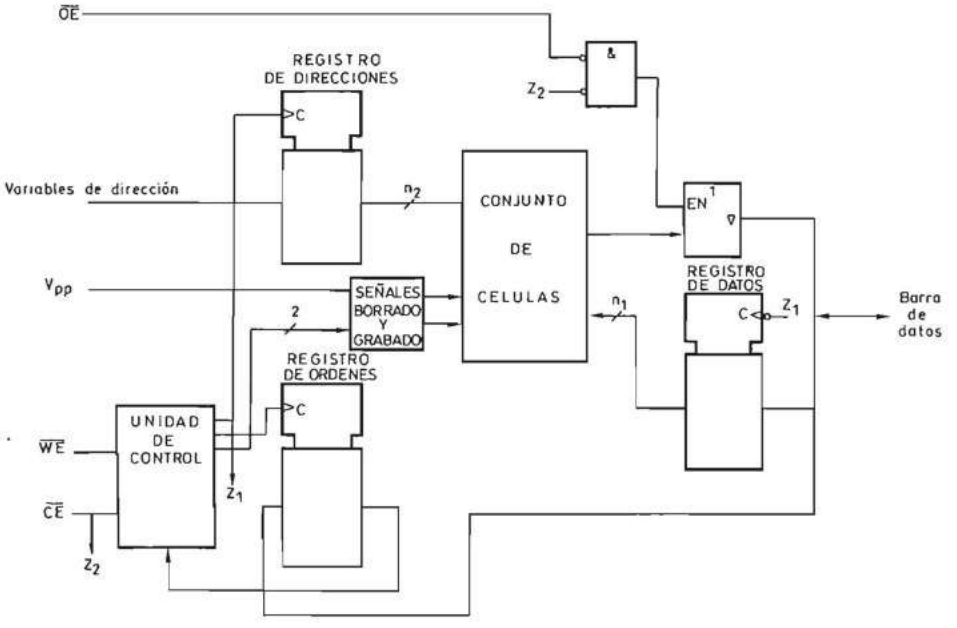


FIGURA 7.56.—**Diagrama de bloques simplificado de una memoria pasiva reprogramable FLASH.**

d) Un registro de órdenes que memoriza en el instante adecuado la combinación procedente de la barra de datos que indica a la unidad de control el proceso que ha de realizar. Las órdenes típicas son:

— Lectura de la memoria (Read memory).

— Lectura del código del tipo de memoria y del fabricante (Intelligent identifier).

— Activación del borrado y acción de borrar (Set-up Erase/Erase).

Se realiza en dos ciclos sucesivos de la señal de escritura ($\overline{WE}$) para evitar borrados accidentales.

— Verificación del borrado (Erase-verify). Tiene como misión verificar que se han borrado todas las posiciones de la memoria.

— Activación del grabado y acción de grabar (Set-up Program/Program).

Se realiza en dos ciclos de la señal $\overline{WE}$. En el primero se introduce en el registro de órdenes la correspondiente al grabado o programación de la memoria y en el segundo se memoriza la dirección y datos a grabar.

— Verificación de la programación (Program-Verify).

Se utiliza para comprobar el resultado de la acción de grabar anteriormente descrita y se debe introducir en el registro de órdenes después de realizar aquélla.

—Reinicialización.

Esta orden tiene como misión la de interrumpir las secuencias de órdenes de borrado o grabación ejecutadas mediante las órdenes correspondientes.

Para un estudio más detallado de las memorias FLASH y su acoplamiento a los microprocesadores se remite al lector a la información de los fabricantes [INTE 90], [SEEQ 87].

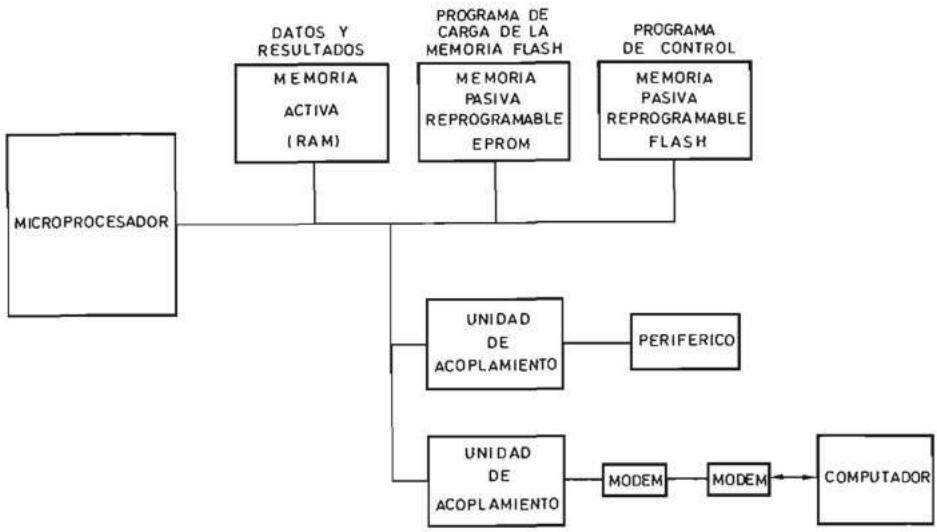


FIGURA 7.57.—**Aplicación de las memorias pasivas reprogramables FLASH al almacenamiento de programas en un sistema de control distribuido.**

En la figura 7.57 se representa una aplicación típica de las memorias FLASH, que está constituida por un microcomputador dedicado al control de un determinado proceso cuyo programa puede ser cambiado a distancia. Para ello la memoria del microcomputador está dividida en tres partes:

— Una memoria activa (RAM) en la cual se almacenan datos y resultados.
— Una memoria pasiva reprogramable FLASH que contiene el programa de control del microcomputador que se desea cambiar en determinados instantes.
— Una memoria pasiva reprogramable EPROM (también puede ser FLASH) que contiene un programa que dialoga con un computador remoto a través de un modem y permite borrar y regrabar el programa contenido en la otra memoria reprogramable.

Un microcomputador con estas características está especialmente orientado al control de procesos en los que se pueden producir situaciones que implican la necesidad del cambio de los algoritmos de control. Un ejemplo típico es el control del tráfico urbano mediante una red de microcomputadores.

7.2.3.1.4. Diseño de unidades de memoria de acceso aleatorio con bloques funcionales. Las memorias de acceso aleatorio de los diversos tipos estudiados en los apartados anteriores se realizan como bloques funcionales en circuito integrado de una cierta capacidad en las diversas tecnologías. Las unidades de memoria de acceso aleatorio se diseñan mediante el acoplamiento adecuado de los bloques funcionales.

A continuación se estudia la forma de acoplar los bloques funcionales para obtener una unidad de memoria. Se supone un bloque funcional como el representado en la figura 7.58 que constituye una memoria de acceso aleatorio activa, estáti-

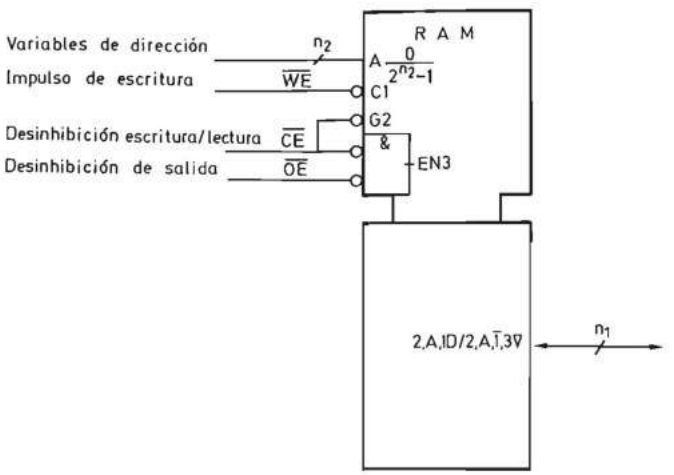


FIGURA 7.58.—**Símbolo lógico de una memoria de acceso aleatorio activa, estática y de escritura y lectura no simultáneas, con terminales de entrada y salida comunes.**

ca y de escritura y lectura no simultáneas, con terminales de entrada y salida comunes. Esta memoria posee 2^{n_2} posiciones de n_1 bits en cada posición.

La forma de conectar los bloques funcionales depende de la relación que existe entre el número de posiciones y el número de bits de cada posición de la unidad de memoria que se desea obtener y los respectivos del bloque funcional utilizado. A continuación se analizan los diferentes casos.

a) Elevación del número de bits de cada posición.

Se supone que el número de posiciones del bloque funcional y de la unidad de memoria coinciden, pero que el número de bits de cada posición de esta última es superior al de aquél.

En la figura 7.59 se representa el esquema obtenido. El número de bloques funcionales necesarios n' es igual al cociente entre el número n'_1 de bits de cada posición de la unidad de memoria y el correspondiente n_1 de cada bloque funcional. Las variables de dirección y las de control $\overline{WE}$, $\overline{CE}$ y $\overline{OE}$ son comunes a todos los bloques funcionales y los terminales de entrada y salida se disponen en paralelo. Si la cargabilidad de salida (fan-out) (medida en unidades de intensidad de corriente en ambos estados lógicos) del sistema que proporciona las variables de dirección y las señales de control es inferior a la del conjunto de bloques funcionales unidos, es necesario el acoplamiento a través de circuitos amplificadores disponibles también como bloques funcionales en circuito integrado.

b) Elevación del número de posiciones.

Se supone que el número de bits de cada posición del bloque funcional y de la unidad de memoria coinciden y que, por el contrario, el número de posiciones de esta última es superior al de aquél. El esquema correspondiente se representa en la figura 7.60.

El número de bloques necesarios es igual al cociente entre el número total de posiciones de la unidad de memoria m y el número de posiciones del bloque funcional 2^{n_2}. En general, m es una potencia de 2, es decir, $m = 2^n$. Por lo tanto, resulta:

$$\frac{m}{2^{n_2}} = \frac{2^n}{2^{n_2}} = 2^{n-n_2} = 2^{n'_2}$$

Para direccionar la unidad de memoria se necesitan n bits tal que $n = n_2 + n'_2$. Los n_2 bits menos significativos se conectan a todos los bloques funcionales y las salidas y entradas de éstos se unen respectivamente entre sí formando una barra de información. En cada instante un solo bloque funcional puede enviar información a la barra en una operación de lectura o recibir la colocada en ella por un sistema externo en una operación de escritura. Por ello las n'_2 variables más significativas de la dirección se conectan a un decodificador de uno entre $2^{n'_2}$ que posee $2^{n'_2}$ variables de salida de las cuales sólo una está activada simultáneamente, la correspondiente a la combinación de las n'_2 variables presente a la entrada. Cada salida de este decodificador se conecta a la señal de control $\overline{CE}$ de cada blo-

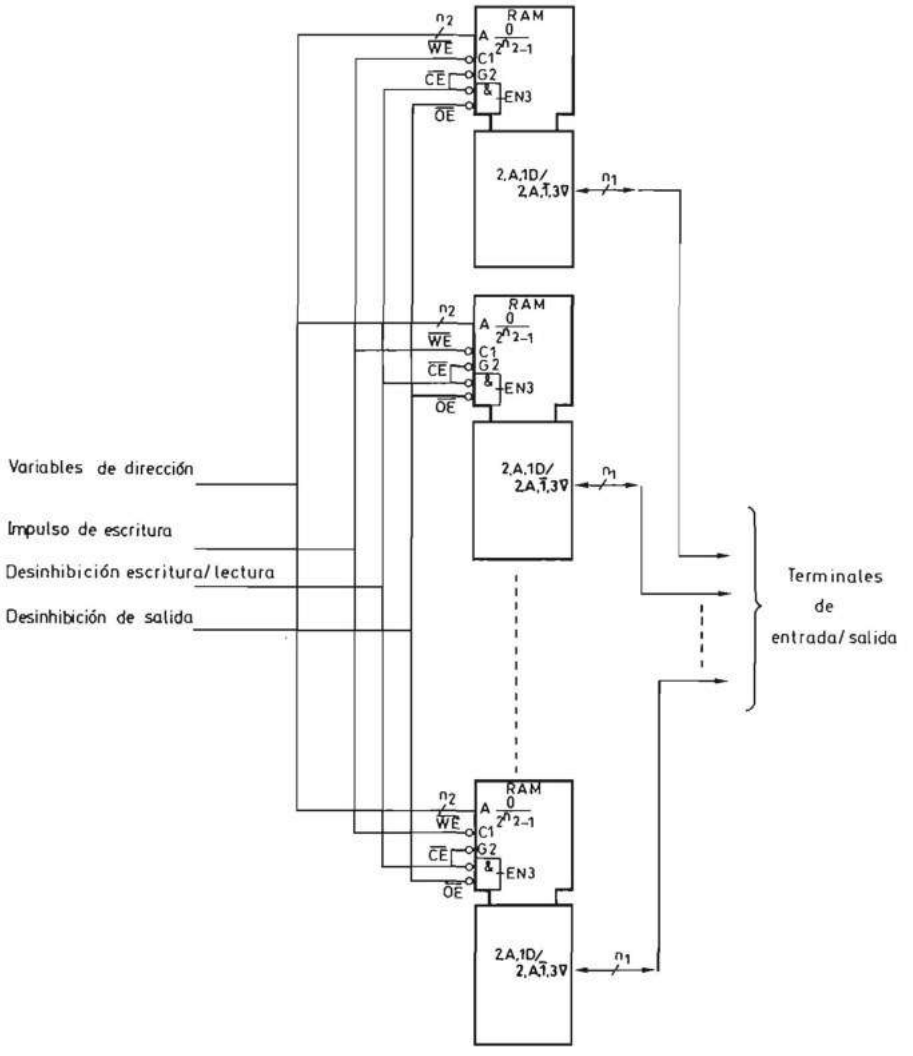


FIGURA 7.59.—**Unidad de memoria de acceso aleatorio activa de 2^{n_2} posiciones de n'_1 bits realizada con n' bloques idénticos al de la figura 7.58.**

que funcional. Para poder inhibir la operación de la unidad de memoria en su totalidad es necesario que el decodificador posea una entrada de inhibición cuya activación desactive todas las $2^{n'_2}$ variables de salida, independientemente de la combinación presente en sus entradas.

Las señales «Impulso de escritura» ($\overline{WE}$) y «Desinhibición de salida» ($\overline{OE}$) son comunes a todos los bloques funcionales. Al igual que en el apartado anterior, si la cargabilidad de salida (fan-out) del circuito que proporciona las señales $\overline{OE}$ y $\overline{WE}$ y las n_2 variables de dirección menos significativas de la dirección, no es

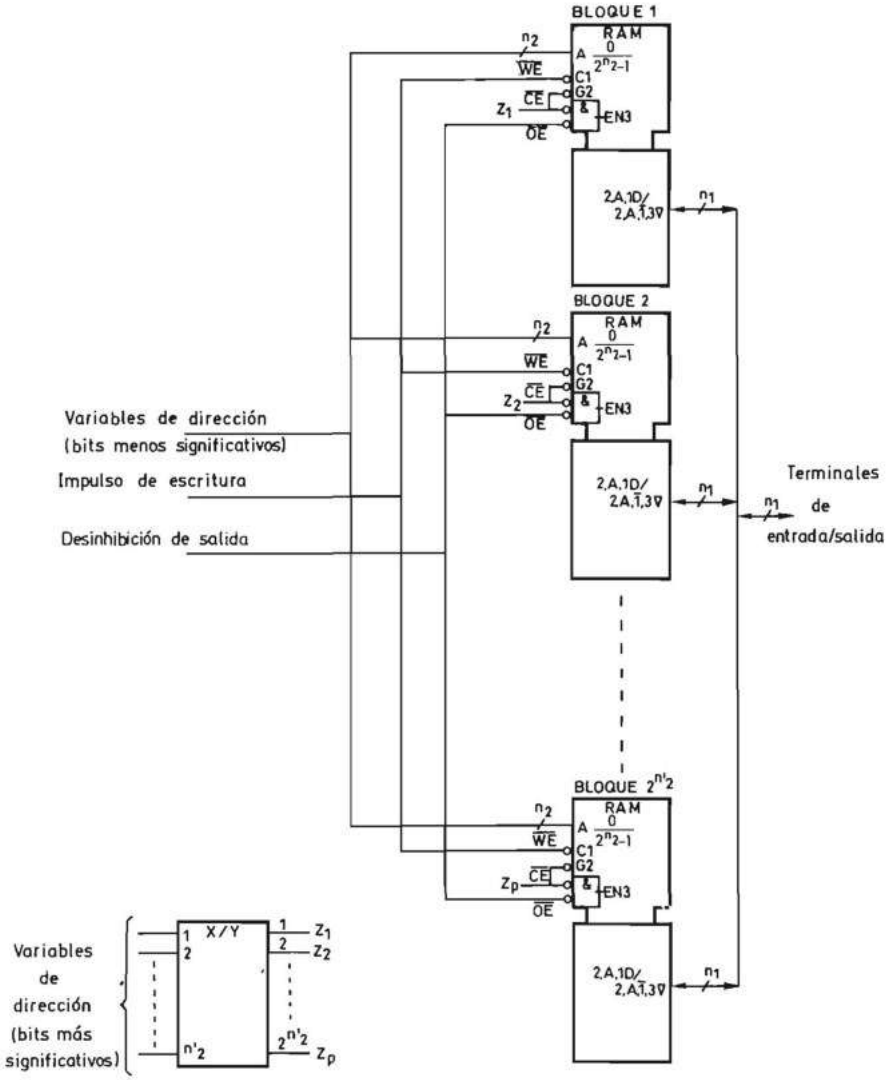


FIGURA 7.60.—**Unidad de memoria de acceso aleatorio activa de n posiciones de n_1 bits realizada con $2^{n'_2}$ bloques idénticos al de la figura 7.58.**

suficiente para realizar la conexión directamente, ésta se efectúa a través de amplificadores disponibles en circuito integrado.

c) Elevación del número de bits de cada posición y del número de posiciones.

Se presenta este caso cuando se desea realizar una unidad de memoria con un número de bits en cada posición y un número de posiciones superior a los del bloque funcional utilizado.

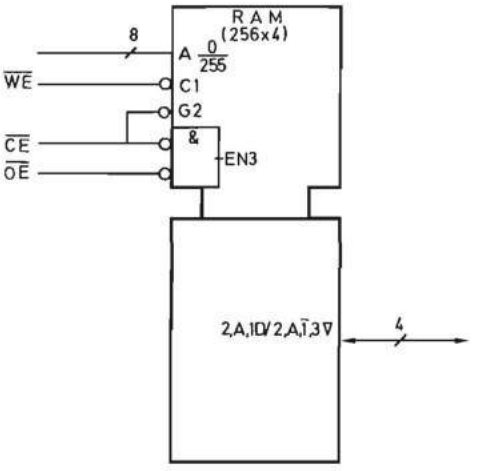


FIGURA 7.61.—**Memoria de acceso aleatorio activa estática de escritura y lectura no simultáneas de 256 × 4.**

Este caso es una combinación de los dos que se acaban estudiar y se analiza mediante un ejemplo práctico.

Ejemplo 7.1: Diseñar una unidad de memoria de acceso de aleatorio de 1 k 8 con un bloque funcional de 256 posiciones de 4 bits (1 k bit) cada una representado en la figura 7.61. Se recomienda al lector que antes de estudiar la solución expuesta a continuación, intente resolver por sí mismo el problema una vez que haya estudiado y comprendido los apartados *a*) y *b*) de párrafos anteriores.

Dado que el número de bits de la unidad de memoria a diseñar es ocho y los del bloque funcional son cuatro, se necesitan dos bloques funcionales con sus terminales de entrada-salida en paralelo. Se obtienen así 256 posiciones de 8 bits cada una. Para lograr 1024 posiciones se han de utilizar por lo tanto 1024/256 = 4 bloques de 256 × 8 tal como se observa en la figura 7.62. Para seleccionar 1024 posiciones se necesitan 10 bits ($2^{10} = 1024$). Los ocho menos significativos (A_0 a A_7) se conectan a todos los bloques funcionales y los otros dos a un decodificador de uno entre cuatro, cuyas salidas se conectan a los terminales de desinhibición de escritura/lectura ($\overline{CE}$) de cada grupo de 256 posiciones de 8 bits constituido por dos bloques funcionales.

Los principios que se acaban de estudiar para la realización de unidades de memoria de acceso aleatorio con bloques funcionales son válidos tanto para las memorias activas (de escritura y lectura) como para las pasivas, y son de gran aplicación en las síntesis de la memoria de datos e instrucciones de los sistemas digitales programables.

7.2.3.2 Memorias de acceso serie. Son aquellas en las que el tiempo que tarda en leerse o en grabarse una posición depende de su situación física en el interior de la memoria. Para introducir información en una posición hay que hacerla pasar

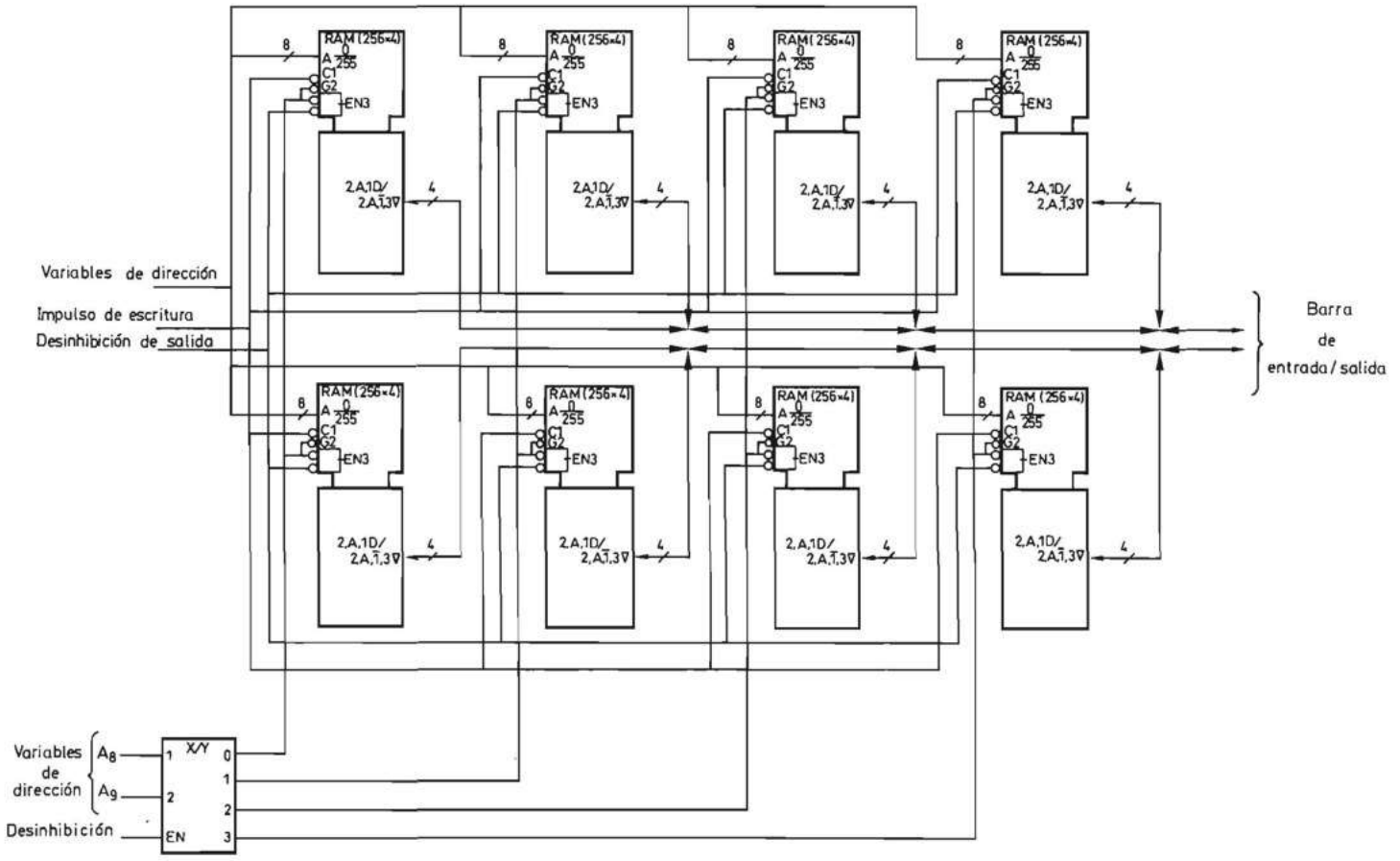


FIGURA 7.62.—**Unidad de memoria de 1k8 realizada con el bloque de la figura 7.61.**

FIGURA 7.63.—**Diagrama de bloques de una memoria de acceso serie.**

por todas las que la preceden a partir de los terminales de entrada. De forma similar, para leer el contenido de una posición hay que hacer pasar su información por todas las que la siguen hasta alcanzar los terminales de salida. De todo lo expuesto se deduce la denominación de acceso serie dada a estas memorias cuyo diagrama de bloques se representa en la figura 7.63.

La información se puede organizar de dos formas principales en las memorias de acceso serie.

a) Organización en serie bit a bit.

En esta organización se colocan en serie las posiciones y los bits de cada posición. El diagrama de bloques de una memoria de este tipo es el representado en la figura 7.64. La memoria posee un terminal único de entrada y otro de salida (el uno se indica de forma explícita para mayor claridad). La memoria posee además uno o más terminales a los cuales se aplican las órdenes de lectura o escritura y que realizan el desplazamiento de la información en el interior de la memoria, por lo que las denominamos órdenes de desplazamiento. En la figura 7.65 se representa la forma en que se dispone la información en una memoria de 2^{n_2} posiciones de n_1 bits cada una organizada en serie bit a bit. El número total de bits de la memoria $2^{n_2} \cdot n_1$.

FIGURA 7.64.—**Diagrama de bloques de una memoria de acceso serie organizada bit a bit.**

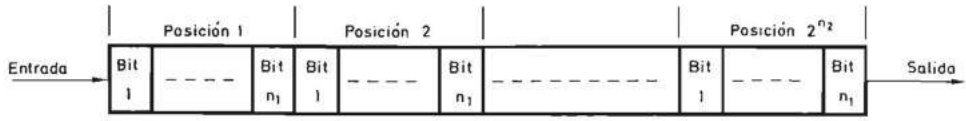


FIGURA 7.65.—**Organización de la información en una memoria de acceso serie bit a bit.**

b) Organización en serie posición a posición.

En esta organización se colocan en serie las posiciones y los bits de cada posición se colocan en paralelo. Por lo tanto, si cada posición posee n_1 bits, una memoria serie organizada de esta forma tiene un diagrama de bloques como el indicado en la figura 7.66, es decir, está constituida por n_1 memorias serie de un único terminal de entrada y otro de salida colocadas en paralelo, con los terminales de órdenes de desplazamiento comunes a todas. En general, el diagrama de bloques de la figura 7.66 se suele representar mediante un bloque único, que es el indicado en la figura 7.67 en el que se indica que la entrada y la salida de información posee n_1 terminales en paralelo.

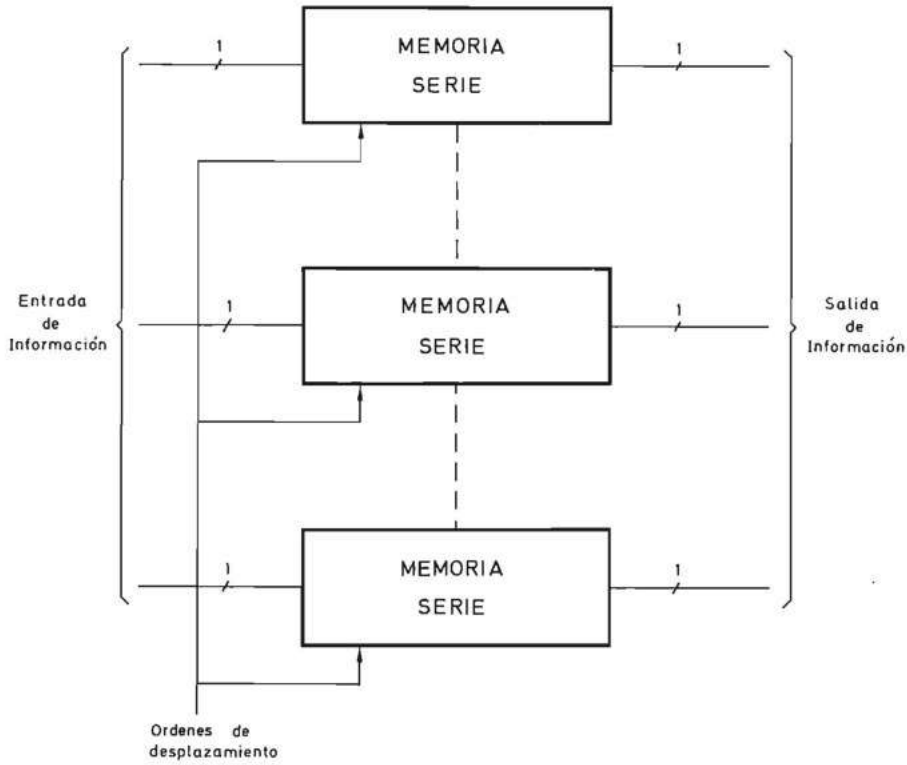


FIGURA 7.66.—**Diagrama de bloques de una memoria de acceso serie con las posiciones organizadas en paralelo.**

FIGURA 7.67.—**Diagrama de bloques simplificado de una memoria de acceso serie con posiciones organizadas en paralelo.**

En la figura 7.68 se representa la forma en que se dispone la información en una memoria de 2^{n_2} posiciones de n_1 bits cada una, organizada en serie posición a posición. Se observa que esta unidad de memoria está contituida por n_1 memorias de 2^{n_2} posiciones de un bit cada una.

Las memorias de acceso serie se pueden clasificar a su vez en tres clases distintas, que se diferencian por la acción que ejercen las órdenes de lectura y escritura externas sobre el desplazamiento de la información en el interior de la memoria. Estudiaremos seguidamente las características y aplicaciones de cada una de estas clases.

7.2.3.2.1 Registros de desplazamiento. Son memorias de acceso serie en las que cada orden externa de escritura o lectura desplaza la información una posición en el interior de la memoria. La orden externa de desplazamiento está constituida por los impulsos de un generador. Si la memoria está organizada en serie posición a posición, para que la información de una posición aparezca en la salida es necesario aplicar al registro tantos impulsos como posiciones existen entre ella y la salida.

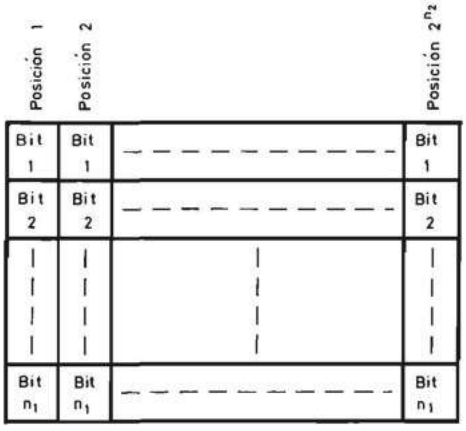


FIGURA 7.68.—**Organización de la información en una memoria de acceso serie con posiciones en paralelo.**

Con los registros de desplazamiento se pueden realizar memorias tanto de acceso directo como de acceso secuencial.

Los registros de desplazamiento se clasifican en dos clases, de acuerdo con las características de frecuencia de los impulsos de desplazamiento:

a) *Registros de desplazamiento estáticos.* Son aquellos en los que los impulsos de desplazamiento pueden anularse por tiempo indefinido. La información queda almacenada en las diferentes posiciones de la memoria a las que fue llevada por el último impulso de desplazamiento aplicado.

Estos registros están constituidos por biestables síncronos activados por flancos conectados en serie, es decir, la entrada de uno está conectada a la salida del anterior y así sucesivamente. Su diagrama de bloques es el representado en la figura 7.69.

FIGURA 7.69.—**Diagrama de bloques de un registro de desplazamiento estático.**

En el apartado 6.3.5.4.4 se realiza un estudio detallado de los registros de desplazamiento estáticos y sus aplicaciones.

Como ampliación a lo allí expuesto se puede indicar que estos registros se utilizan, junto con otros circuitos digitales, para facilitar el intercambio de información entre varios procesadores o entre un procesador y sus periféricos. En la mayoría de los casos los procesadores trabajan con un cierto número n de bits en paralelo y por el contrario el intercambio de información entre procesadores situados a una cierta distancia se realiza en serie bit a bit. Por otra parte, la comunicación entre procesadores independientes precisa una sincronización a través del establecimiento de un protocolo.

Por todo ello se crea el concepto de procesador de comunicaciones que, en general, se divide en dos partes: una dedicada a la transmisión y otra a la recepción, tal como se representa esquemáticamente en la figura 7.70. El elemento básico del transmisor es un registro de desplazamiento estático de entrada en paralelo y salida serie, y el del receptor un registro de desplazamiento estático de entrada serie y salida paralelo.

b) *Registros de desplazamiento dinámicos.* Son aquellos en los que los impulsos de desplazamiento no pueden anularse por tiempo indefinido porque en este caso desaparece la información contenida en su interior. Se han realizado estos re-

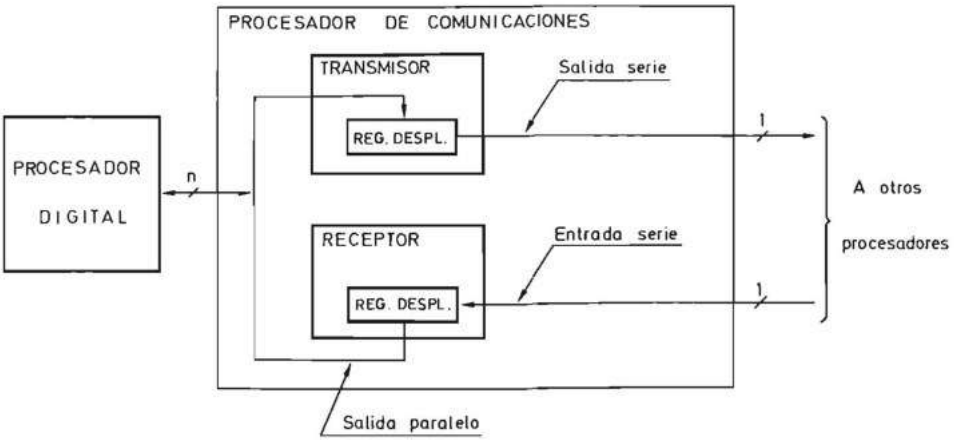


FIGURA 7.70.—**Aplicación de los registros de desplazamiento estáticos en la realización de un procesador de comunicaciones.**

gistros especialmente en tecnologías MOS y, al igual que las memorias de acceso aleatorio activas dinámicas, están formados por células básicas de memoria que almacenan la información en forma de carga del condensador existente entre la puerta y el surtidor de un transistor MOS. En el apartado 7.3.3 se describen algunos circuitos de registros de desplazamiento dinámicos.

La sencillez de las células básicas utilizadas en este registro ha permitido su realización como bloques funcionales en circuitos integrados de gran escala de integración en una época (década de 1960) en la que todavía no existían las memorias de acceso aleatorio activas estáticas o dinámicas. Por ello han sido utilizadas extensamente como memorias de acceso directo en la realización de procesadores digitales que reciben y entregan información en serie como son las calculadoras de bolsillo o sobremesa. Aunque esta aplicación ha dejado de tener interés práctico debido al desarrollo de las memorias de acceso aleatorio activas, su estudio tiene un gran interés didáctico, dadas las características de operación de los circuitos que utilizan.

Por ello a continuación se analizan los circuitos que han de añadirse a un registro de desplazamiento dinámico para convertirlo en una memoria de acceso directo.

La característica de que la información contenida en un registro de desplazamiento dinámico se pierda si dejan de aplicarse impulsos de desplazamiento durante más de un cierto tiempo hace que en funcionamiento normal (sin realizar ninguna operación de escritura o lectura) la salida tenga que estar conectada a la entrada y la información recircule de forma continuada en el interior del registro. Por lo tanto, el esquema adecuado para que se produzca la recirculación es el indicado en la figura 7.71. Si se desea seleccionar una posición de memoria para introducir información en ella o leer la que contiene, es necesario conocer cuándo está situada en los terminales de salida. Esto se consigue contando los impulsos del generador mediante un sistema contador síncrono descrito en el apartado 6.3.5.4.3.3. La

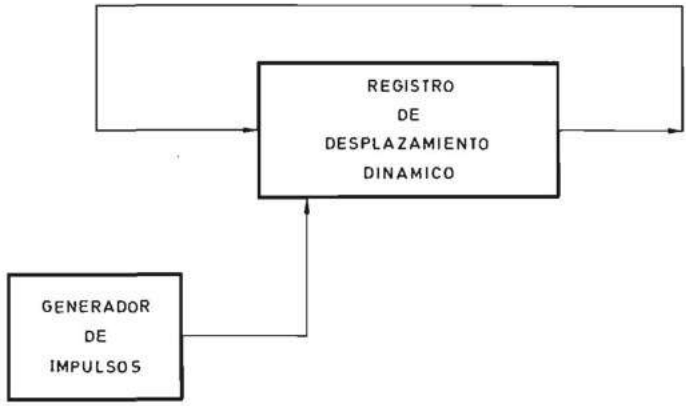


FIGURA 7.71.—**Esquema básico de un registro de desplazamiento dinámico utilizado como memoria de acceso directo.**

capacidad de este contador depende del número de posiciones del registro y de la forma en que esté organizado.

Supongamos que el registro posee 2^{n_2} posiciones de n_1 bits. Si la información está organizada en serie posición a posición, el contador ha de poseer 2^{n_2} estados y, por lo tanto, ha de poseer n_2 biestables, tal como se indica en la figura 7.72. Cada vez que este contador llegue al estado cero, indica que la primera posición se encuentra en el terminal de salida. Si la información está organizada en serie bit a bit, el contador ha de tener $2^{n_2} \cdot n_1$ estados, tal como se indica en la figura 7.73. El contador está dividido en dos, conectados entre sí, constituyendo el conjunto un contador síncrono: un contador de n_1 estados que realiza el contaje de

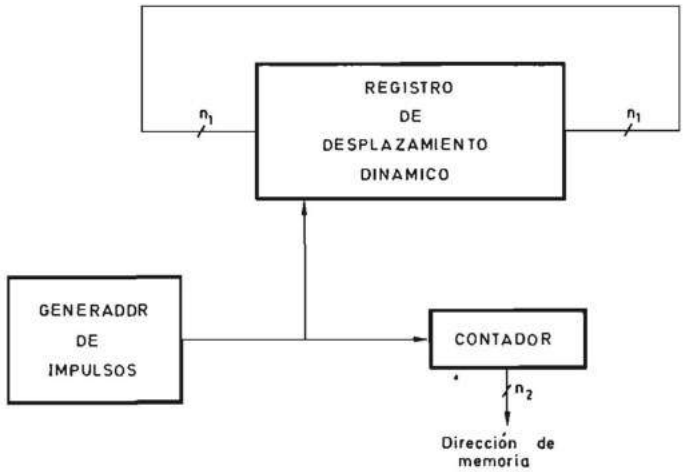


FIGURA 7.72.—**Esquema de una memoria de acceso directo realizada con un registro de desplazamiento dinámico con direccionamiento mediante un contador.**

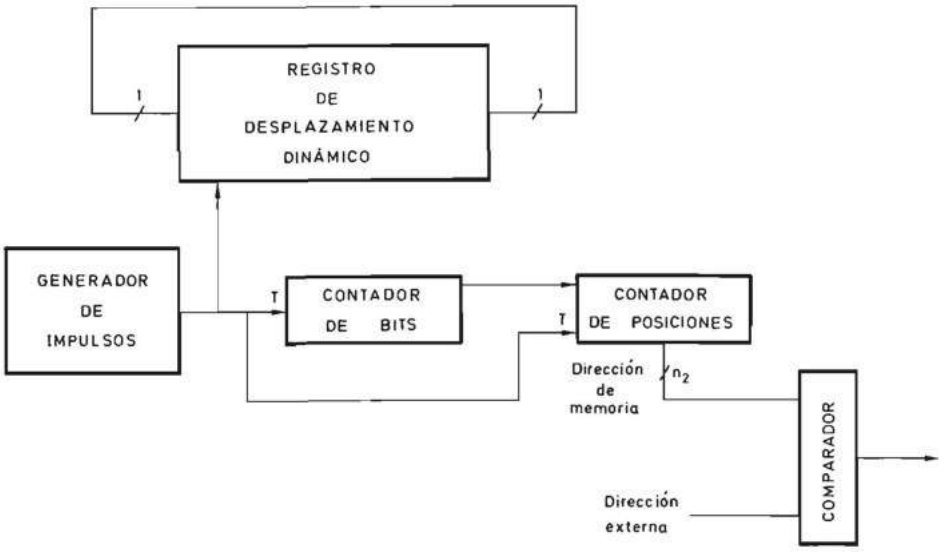


FIGURA 7.73.—**Esquema de una memoria de acceso directo realizada con un registro de desplazamiento dinámico con circuito de selección de posiciones.**

los bits de cada posición y un contador de 2^{n_2} estados (n_2 biestables) que incrementa en uno su contenido por cada ciclo completo del contador de bits y que recibe el nombre de contador de posiciones. El estado de este contador indica la posición del registro que está saliendo por el terminal de salida en cada instante. La forma más sencilla de seleccionar la posición que se desea leer o escribir es comparar la combinación binaria que le corresponde con el estado del contador. La salida del comparador permanecerá en un estado activo durante todo el tiempo que los n_1 bits de la posición seleccionada se encuentran en el terminal de salida (figura 7.73).

Veremos seguidamente la forma de realizar la lectura del contenido de una posición de un registro organizado en serie bit a bit. Se utiliza para ello un registro de desplazamiento estático de n_1 bits cuya entrada serie se conecta a la salida del registro de desplazamiento dinámico. En la figura 7.74 se representa el circuito correspondiente. La entrada T del registro estático recibe los impulsos del mismo generador que desplaza la información en el registro dinámico. Se supone que el registro de desplazamiento posee una entrada de inhibición que, cuando se encuentra en estado cero, impide el desplazamiento aunque se apliquen impulsos en la entrada T. (Por ejemplo, si los biestables utilizados son J-K, cuando la entrada de inhibición se encuentre en estado cero pondrá en estado cero ambas entradas). La entrada de inhibición se controla mediante el producto lógico de la orden de lectura que pasa al nivel uno cuando se desea realizar esta operación, y la salida del comparador que adopta dicho estado cuando coinciden la combinación binaria correspondiente a la posición que se desea leer y el contenido del contador de posiciones. Los n_1 impulsos del generador que se aplican a la entrada T mientras la salida

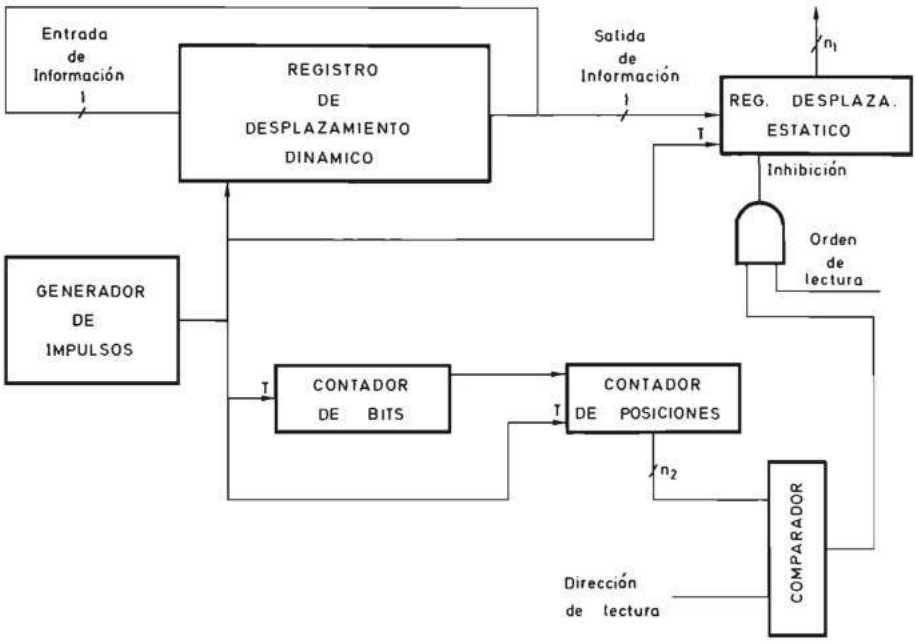


FIGURA 7.74.—**Esquema del circuito de lectura de una memoria de acceso directo realizada con un registro de desplazamiento dinámico.**

del comparador se encuentra en estado uno, introducen en serie en el registro estático la información de la posición que se deseaba leer. Simultáneamente la información recircula en el registro dinámico y, por lo tanto, la lectura no es destructiva.

En la figura 7.75 se representa el circuito para introducir información en una posición seleccionada mediante una combinación binaria aplicada al comparador. La información que se desea escribir en la posición seleccionada se introduce en paralelo en un registro de desplazamiento estático. La salida del registro de desplazamiento dinámico no se conecta directamente a la entrada sino a través de un multiplexor de dos canales. El otro canal del multiplexor se conecta a la salida serie del registro de desplazamiento estático. Cuando no se da una orden de escritura, la variable de selección del multiplexor se encuentra en estado lógico cero y la información recircula. Cuando la orden de escritura pasa al nivel lógico uno, la entrada de inhibición del registro de desplazamiento estático y la variable de selección del multiplexor pasan al nivel lógico uno durante el tiempo que el comparador se encuentra en dicho estado. De esta forma se corta la recirculación y la información contenida en el registro de desplazamiento estático se introduce en serie en el registro de desplazamiento dinámico. La orden de escritura sólo debe permanecer en estado uno durante un ciclo completo de recirculación del registro de desplazamiento dinámico. De esto se encarga la unidad de control que genera todas las señales necesarias para realizar las operaciones de escritura y lectura. La combinación de las figuras 7.74 y 7.75 permite la lectura o escritura de cualquier posición.

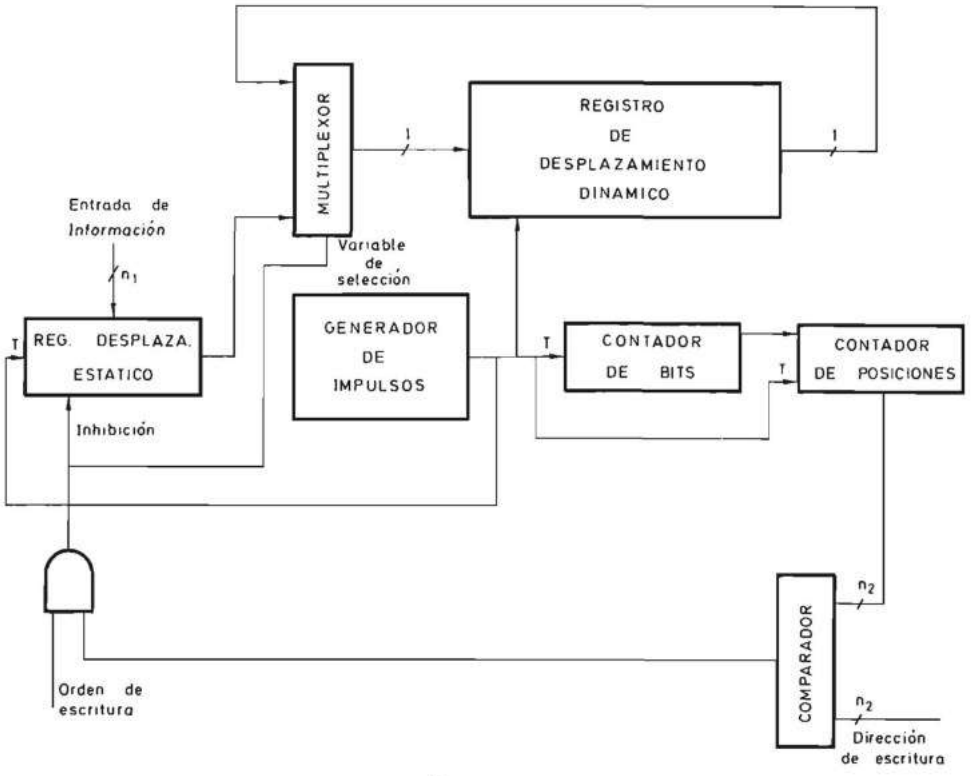


FIGURA 7.75.—**Esquema del circuito de escritura en una memoria de acceso directo realizada con un registro de desplazamiento dinámico.**

En muchas aplicaciones en las que no es necesario seleccionar la posición en que se desea escribir la información se introduce ésta en la posición cero desplazando una posición toda la información contenida en el registro dinámico. En la figura 7.76 se representa el circuito adecuado para realizar dicha transferencia. El registro de desplazamiento estático en el que se introduce en paralelo la información a escribir en el registro de desplazamiento dinámico tiene su entrada unida a la salida de éste. Su salida está unida a una de las entradas del multiplexor cuya salida está a su vez conectada a la entrada del registro de desplazamiento dinámico.

Cuando la unidad de control recibe una orden externa de escritura, espera hasta recibir indicación del paso por el estado cero del contador de posiciones. A partir de ese instante genera la orden de escritura que pasa al estado uno durante un ciclo completo del contador de posiciones. De esta forma se desinhibe el registro de desplazamiento estático y la variable de selección del multiplexor hace que se conecte la salida del registro de desplazamiento estático a la entrada del dinámico. Así se logra que la información externa entre en la primera posición del registro dinámico y la información que estaba situada en ésta se desplace a la segunda y

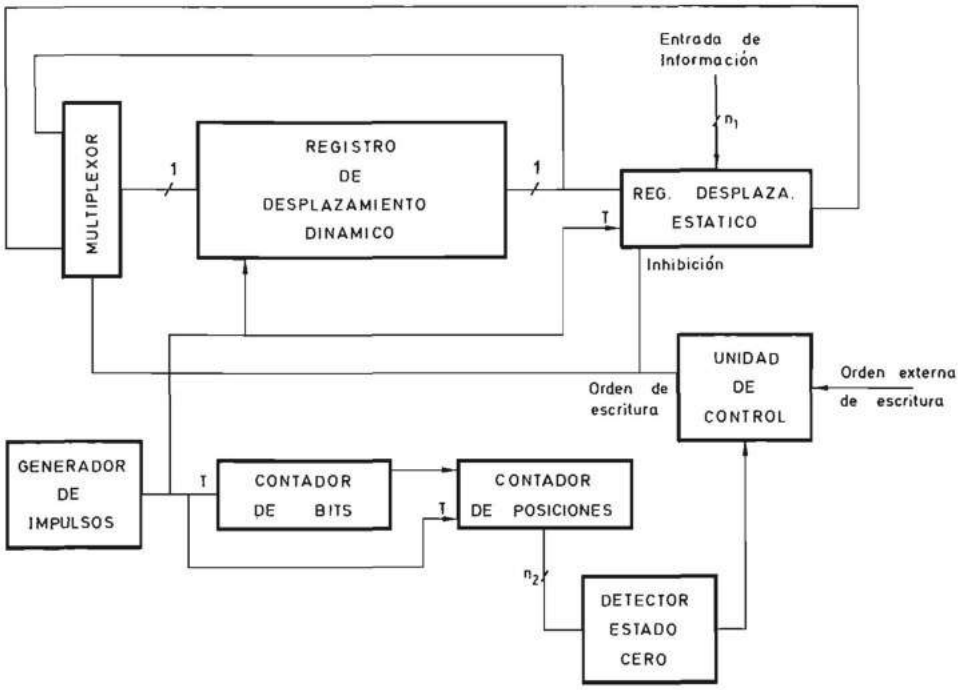


FIGURA 7.76.—**Esquema del circuito de escritura secuencial en un registro de desplazamiento dinámico.**

así sucesivamente. Al terminar el ciclo desaparece la orden de escritura y el registro dinámico vuelve a recircular normalmente.

Uno de los ejemplos más característicos de aplicación de los registros de desplazamiento dinámico son las calculadoras con teclado accionado manualmente por un operador. En la figura 7.77 se representa el diagrama de bloques de una calculadora que describiremos brevemente a continuación.

El accionamiento de algún pulsador del teclado actúa sobre la unidad de control y hace que ésta genere señales distintas según cual sea el pulsador que se haya activado.

Cuando se ha pulsado una tecla numérica, la unidad de control introduce en el registro de entrada su equivalente binario y a continuación pasa el contenido de éste al registro de desplazamiento dinámico a través del multiplexor de una forma similar a la descrita en la figura 7.76. Cuando se pulsa una tecla operativa cualquiera del teclado, la unidad de control genera las señales necesarias para que operen entre sí los contenidos de dos posiciones del registro de desplazamiento dinámico por medio de una unidad aritmética y se almacene el resultado en una de ellas. El biestable auxiliar almacena el contenido de una posición temporalmente para que opere con el contenido de otra posición y almacenar el resultado en esta última. El registro de salida almacena la información a visualizar. La visualización se

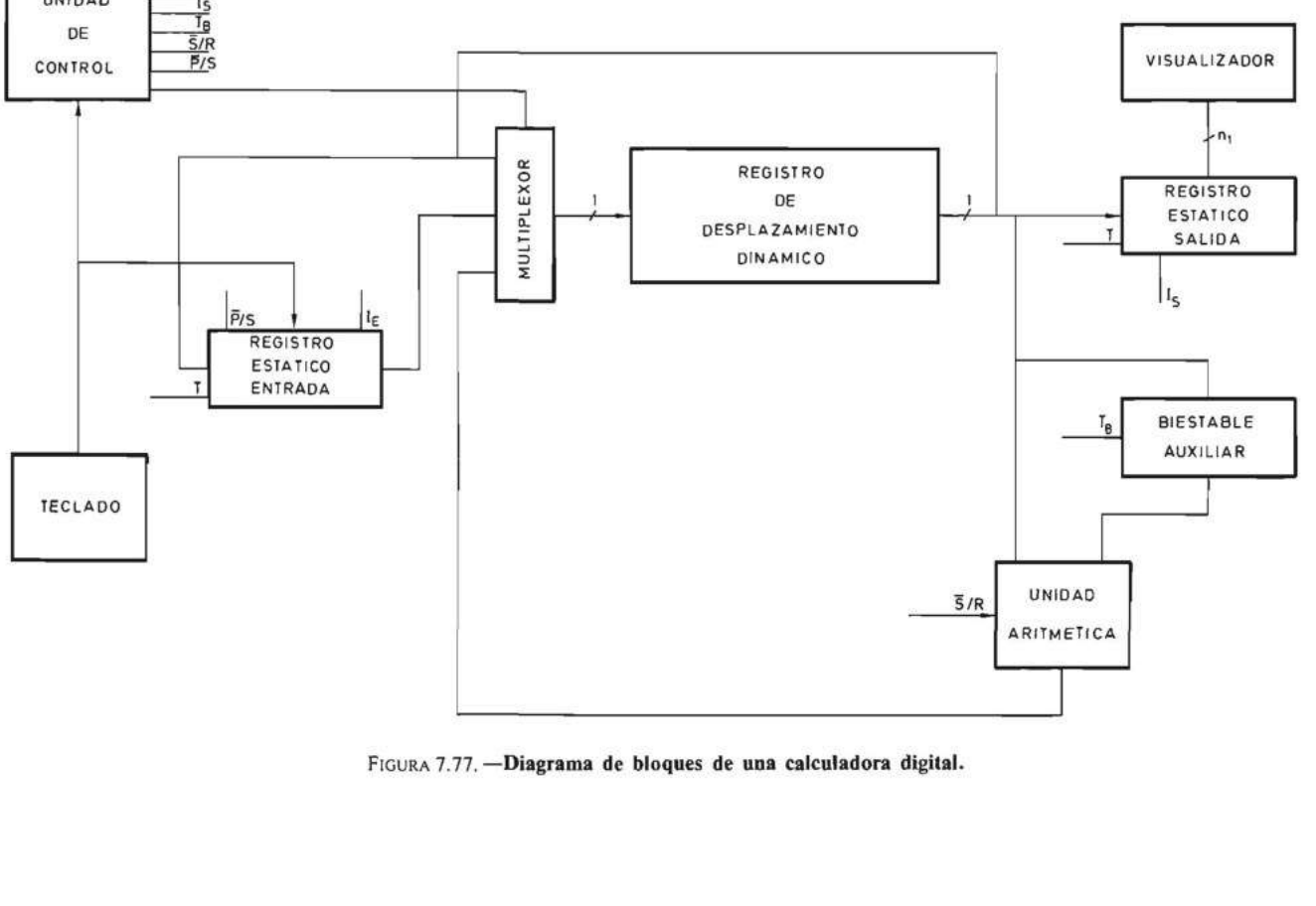


FIGURA 7.77. —**Diagrama de bloques de una calculadora digital.**

hace de forma dinámica, es decir, los diferentes dígitos se activan en secuencia y el observador los ve todos activados debido a la inercia de la retina. La visualización dinámica es muy adecuada para la salida serie.

La utilización masiva de las calculadoras ha hecho que el sistema de la figura 7.77 haya sido realizado en un único circuito integrado de gran escala de integración.

7.2.3.2.2 Memorias TUBO (FIFO). Son memorias de acceso serie en las que la primera información que entra es la primera en salir [en inglés «First In-First Out» (FIFO)]. En la figura 7.78 se representa el diagrama de bloques de una memoria TUBO organizada en serie posición a posición, que posee n_1 terminales de entrada y n_1 terminales de salida y un conjunto de terminales de control que dependen de la forma en que se implementa la memoria. También se las denomina memoria COLA.

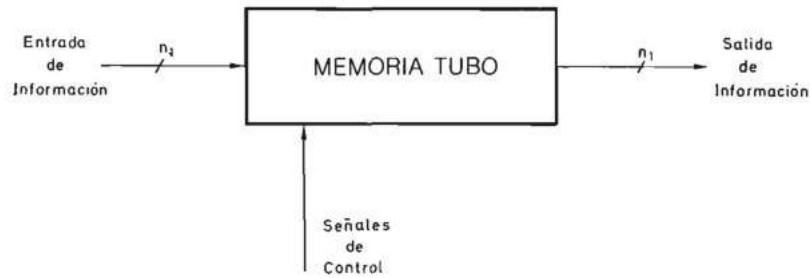


FIGURA 7.78.—**Diagrama de bloques de una memoria TUBO (FIFO).**

Las memorias TUBO se pueden implementar de dos formas diferentes:

a) Mediante registros de desplazamiento en los que la información se desplaza a través de sus diferentes posiciones.
b) Mediante memorias de acceso aleatorio en las que la información no se mueve y lo que cambia es el contenido de unos registros que direccionan la memoria.

A continuación se analizan cada una de ambas implementaciones.

a) Implementación mediante un registro de desplazamiento.

La forma en que se mueve la información en el interior de una memoria TUBO implementada mediante un registro de desplazamiento, al realizar sucesivas operaciones de escritura y lectura, se representa en la figura 7.79.

En la parte superior se representa la memoria vacía en el instante inicial antes de realizar ninguna operación de escritura o lectura. Si se supone que a continuación se da una orden de escritura, la información I_1 presente a la entrada debe desplazarse hasta la última posición de la memoria conectada a la salida. Si a continuación se realiza otra operación de escritura, la información I_2 presente en la

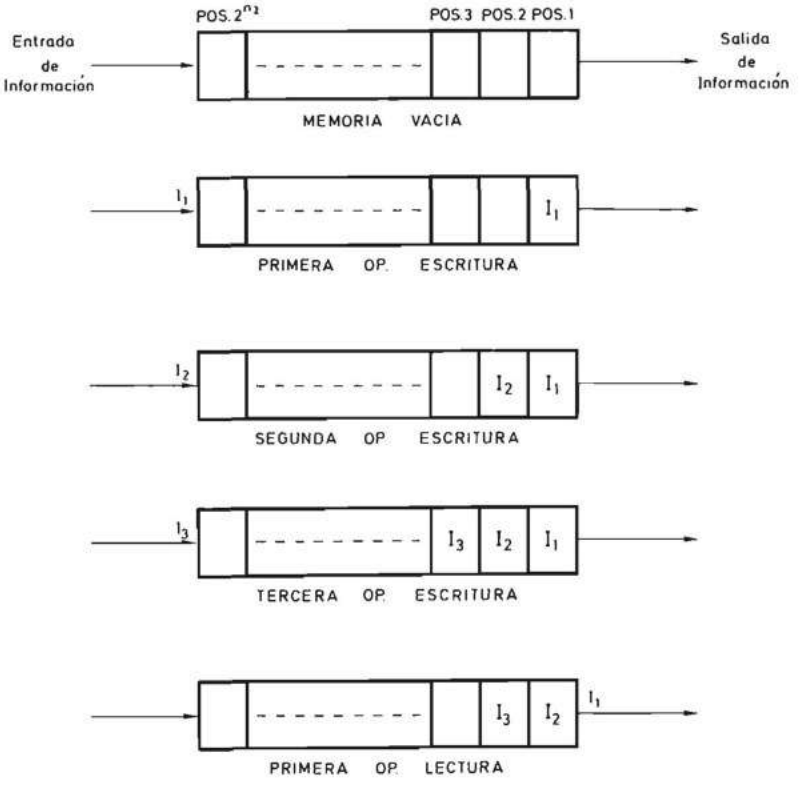


FIGURA 7.79.—**Forma de almacenamiento de la información en sucesivas operaciones de escritura y lectura de una memoria TUBO (FIFO).**

entrada se desplaza hasta la posición vacía más próxima a la salida, que es ahora la penúltima. Se supone que a continuación se realiza otra operación de escritura; la información I_3 presente en la entrada se desplaza también hasta la posición vacía más cercana a la salida que es la antepenúltima. Al realizar seguidamente una operación de lectura, la información contenida en la memoria se desplaza una posición hacia la salida, es decir, la información I_1 sale de la memoria TUBO (FIFO), la información I_2 se desplaza a la posición ocupada por I_1, y la información I_3 se desplaza a la que contenía a I_2. De todo lo expuesto el lector puede deducir que la memoria TUBO (FIFO) se comporta igual que un registro de desplazamiento en la operación de lectura, pero no así en la operación de escritura, en la que la información se desplaza desde la entrada hasta la posición vacía más cercana a la salida.

Una memoria TUBO puede realizarse con un registro de desplazamiento estático y una unidad de control que es un sistema secuencial síncrono (fig. 7.80). Esta unidad de control recibe las órdenes externas, que pueden consistir en flancos de subida o bajada; simultáneamente se aplica una señal denominada «Nivel de escri-

tura/lectura». Los niveles cero o uno de esta señal en el instante en que se aplica la orden externa a la unidad de control hacen que se realice la operación de escritura o la de lectura. La unidad de control genera las 2^{n_2} señales correspondientes a las entradas T de los biestables de las diferentes posiciones de la memoria TUBO. En una operación de escritura se aplican a todas las posiciones vacías un número de impulsos de desplazamiento igual precisamente al número de aquéllas para hacer que la información situada a la entrada se desplace a la posición vacía más cercana a la salida. Después de aplicar el primer impulso de desplazamiento, la unidad de control debe colocar un cero a la entrada de la primera posición para que todas las posiciones anteriores a la última vacía queden vacías. Esto se logra mediante una puerta Y tal como se indica en la figura 7.80. En una operación de lectura se aplica un impulso de desplazamiento a todas las posiciones de la memoria.

La unidad de control ha de conocer en todo instante el número de posiciones vacías u ocupadas de la memoria. Esto puede realizarse mediante un contador reversible que cuenta en sentido ascendente las órdenes de escritura y en sentido descendente las de lectura; el contenido de este contador indica continuamente el número de posiciones de la memoria ocupadas.

Las memorias TUBO implementadas mediante registros de desplazamiento poseen las siguientes características que limitan su operatividad:

— En ellas la información se desplaza físicamente lo cual hace que su escritura necesite tantos impulsos del reloj como posiciones vacías posea la memoria.
— La elevación de su número de posiciones disminuye su velocidad porque aumenta el tiempo medio de escritura.

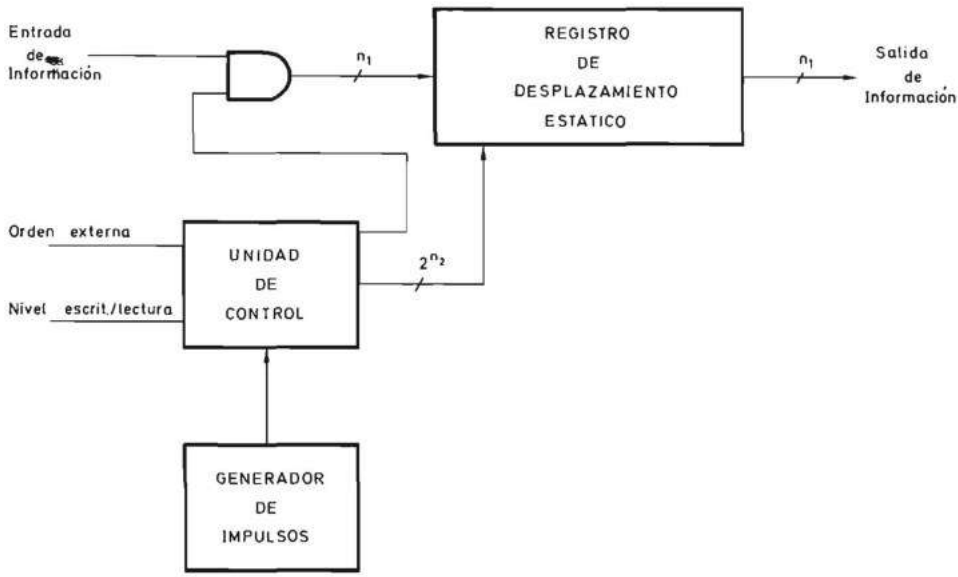


FIGURA 7.80.—**Memoria TUBO (FIFO) realizada mediante un registro de desplazamiento estático y un sistema secuencial síncrono.**

— No es posible realizar en ellas operaciones de lectura y escritura simultáneas.
— No es posible realizar dos lecturas sucesivas de la misma posición.

Estos inconvenientes han hecho que las memorias TUBO hayan dejado de implementarse mediante registros de desplazamiento en el instante en que los fabricantes de circuitos integrados alcanzaron el nivel de muy gran escala de integración (VLSI) y, como consecuencia de ello, fueron capaces de hacerlo mediante memorias de acceso aleatorio combinadas con una unidad de control síncrona. Por ello no se considera conveniente profundizar más en el estudio de las memorias TUBO realizadas mediante registros de desplazamiento.

b) Implementación mediante una memoria de acceso aleatorio.

Uno de los tipos de memorias de acceso aleatorio activas más adecuado para implementar una memoria TUBO es la memoria de escritura y lectura simultáneas en dos posiciones diferentes cuyo diagrama de bloques se representa en la figura 7.23.

En la figura 7.81 se representa su diagrama de bloques constituido por un procesador digital secuencial síncrono y dos contadores, uno de escritura conectado a los terminales de dirección de la memoria que seleccionan la posición en la cual se escribe y otro de lectura conectado a los terminales de dirección de la memoria que seleccionan la posición cuyo contenido se lee. Las operaciones de escritura hacen que el contenido del contador de escritura se incremente en una unidad y otro tanto sucede con las operaciones de lectura y el contador de lectura. La diferencia entre el contenido de ambos contadores (realizada por el procesador) indica el número de posiciones vacías que posee la memoria.

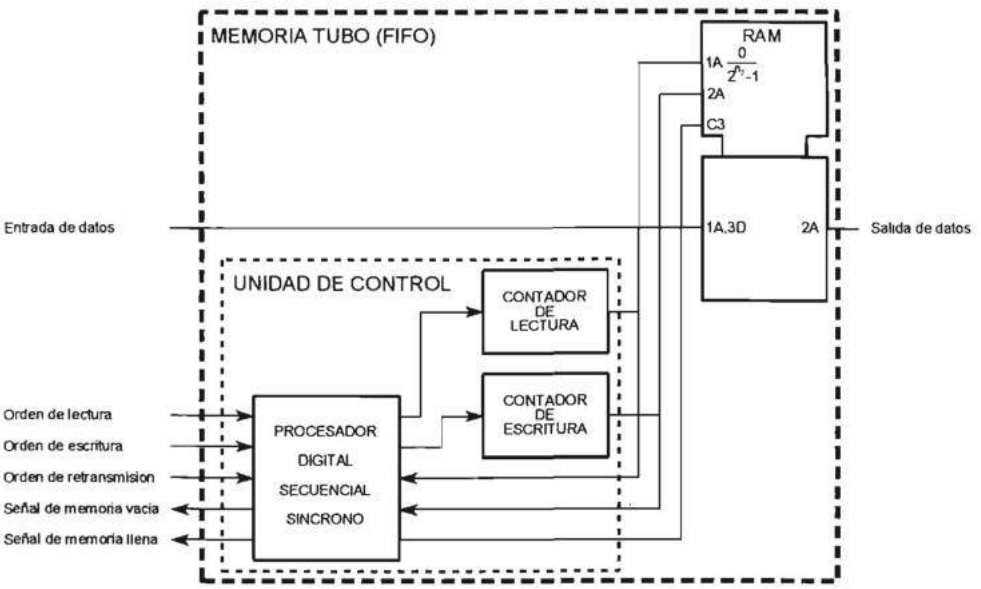


FIGURA 7.81

Esta forma de implementación hace que la relación entre la escritura y la lectura de la información en la memoria sea idéntica a la indicada en la figura 7.79 para la implementación basada en un registro de desplazamiento, pero se diferencia de ella porque posee las siguientes características:

- Todas las operaciones de escritura tienen idéntica duración independientemente del número de posiciones vacías que posea la memoria y de la capacidad de la misma.
- Es posible realizar la operación de escritura y la de lectura simultáneamente debido a que la memoria de acceso aleatorio es de escritura y lectura simultáneas.
- Se puede ampliar la capacidad de la memoria sin disminuir la velocidad. Para ello sólo hay que elevar adecuadamente la capacidad de los dos contadores de la unidad de control.
- Es posible realizar varias operaciones de lectura sucesivas de una o más posiciones de la memoria consecutivas. Para ello se dota a la memoria de un terminal que hace que el contador de lectura se decremente en una unidad. Dicho terminal suele denominarse orden de retransmisión porque su principal utilidad es hacer que se vuelva a enviar la información contenida en la memoria cuando la misma forma parte de un procesador de comunicaciones en la aplicación descrita a continuación.
- Existen múltiples alternativas que se diferencian por las señales de control que la acoplan con los sistemas externos. En la figura 7.81 se representa una configuración típica en la que las señales de control son:

  - Una orden de lectura que consiste en un impulso que un sistema externo aplica a la memoria TUBO para que realice una operación de lectura de la posición seleccionada mediante el contenido del contador de lectura.
  - Una orden de escritura que consiste en un impulso que un sistema externo aplica a la memoria TUBO para que realice una operación de escritura en la posición seleccionada mediante el contenido del contador de escritura.
  - Una orden de retransmisión que consiste en un impulso que un sistema externo aplica a la memoria para que realice una operación de lectura del contenido de la posición leída en la última operación de lectura efectuada.
  - Una señal de memoria llena que se activa cuando el número de operaciones de escritura sucesivas, sin haber realizado ninguna de lectura, iguala al número de posiciones de la memoria de acceso aleatorio.
  - Una señal de memoria vacía que se activa cuando el número de operaciones de lectura sucesivas iguala al de operaciones de escritura con lo cual el contenido de la memoria es nulo.

Para comprender la funcionalidad de las señales de control de la memoria TUBO es necesario analizar su principal aplicación que es la de servir de almacén temporal de información entre dos procesadores digitales síncronos en aquellos casos en los que uno necesita recibir una gran cantidad de información procedente del otro y se quiere hacer la transferencia sin que ningún procesador tenga que estar ex-

FIGURA 7.82.—**Esquema básico del acoplamiento entre dos procesadores digitales síncronos realizados mediante una memoria TUBO.**

cesivamente pendiente del otro. Tal como se indica en la figura 7.82 la memoria TUBO forma parte de la unidad de acoplamiento entre ambos. Un ejemplo típico es el acoplamiento de un procesador digital síncrono a una línea o a un sistema de comunicaciones (como por ejemplo una línea punto a punto, una red de área local, etc.) realizado mediante un sistema digital que posee una unidad de comunicaciones que lleva a cabo todas las tareas de proceso de información necesarias para realizar la transmisión de información. Dicho sistema suele recibir la denominación de procesador de comunicaciones (figura 7.83) y su existencia, especialmente la inclusión de una memoria TUBO en su interior, constituye actualmente, debido al progreso de la microelectrónica, la mejor solución para descargar al procesador digital síncrono de las tareas de comunicación y permitir que se centre en la ejecución del proceso que tiene que realizar en la aplicación concreta de que se trate. En la figura 7.84 se indica una forma de conexión de la memoria TUBO de la figura 7.81 al procesador digital síncrono por una parte y a la unidad de comunicaciones por otra. Una forma posible de operación es la siguiente:

a) El procesador digital llena la memoria mediante la aplicación de sucesivas órdenes de escritura.

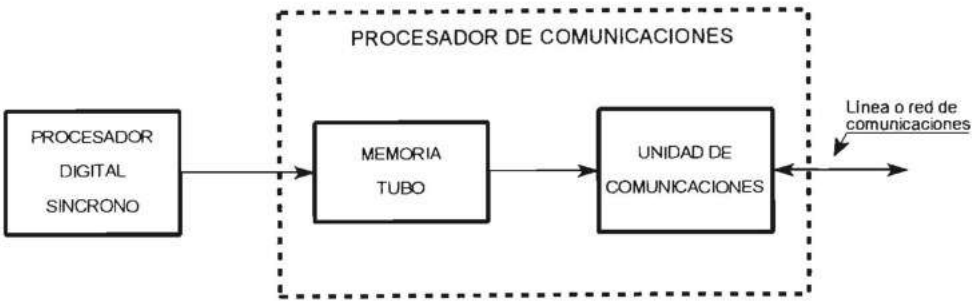


FIGURA 7.83.—**Esquema del acoplamiento de un procesador digital síncrono a una línea o red de comunicaciones por medio de un procesador de comunicaciones que posee una memoria TUBO (FIFO).**

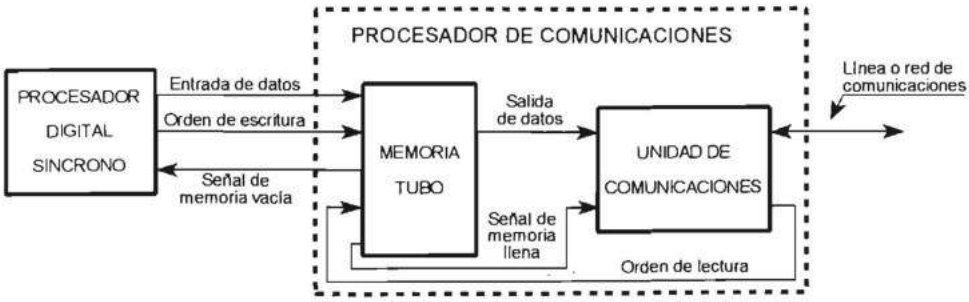


FIGURA 7.84.—**Conexión de una memoria TUBO situada en un procesador de comunicaciones.**

b) La unidad de comunicaciones observa la señal de memoria llena y al detectar que se activa procede a la lectura de todas las posiciones de la memoria mediante la aplicación de sucesivas órdenes de lectura.
c) Si la unidad de comunicaciones detecta que se han producido errores en la transmisión, procede a realizar una lectura de la memoria TUBO mediante la aplicación de la oportuna orden de retransmisión.
d) Cuando el procesador digital detecta la activación de la señal de memoria vacía procede a llenarla de nuevo.

7.2.3.2.3 Memorias PILA (LIFO). Son memorias de acceso serie en las que la información que entra en la última operación de escritura es la que sale en la primera operación de lectura que se realice (last in-first out). En la figura 7.85 se indica gráficamente la forma en que se realiza la escritura y la lectura de una memoria PILA (LIFO). Se supone que inicialmente la memoria está vacía (parte superior de la figura). Los terminales de entrada y de salida corresponden a la misma posición de la memoria. La primera operación de escritura introduce la información I_1 presente en los terminales de entrada en la primera posición de la memoria. Se supone que a continuación se da una segunda orden de escritura que introduce la información I_2 presente en la entrada en la primera posición y traslada I_1 a la segunda posición. Seguidamente se realiza una tercera operación de escritura que introduce la información externa en la primera posición, y traslada I_2 a la segunda posición e I_1 a la tercera. A continuación se realiza una operación de lectura que transfiere la información de la primera posición I_3 al exterior y traslada I_2 a la primera posición e I_1 a la segunda. El lector puede deducir que en una operación de escritura toda la información contenida en la memoria se desplaza una posición hacia la derecha y en una operación de lectura se desplaza una posición hacia la izquierda.

En la figura 7.86 se representa el diagrama de bloques de una memoria PILA organizada en serie posición a posición de n_1 bits de capacidad. La memoria posee n_1 terminales de entrada de información, otros tantos de salida de información y las señales de control. En general, las memorias PILA reciben información y la entregan al mismo sistema digital. Por ello las señales de control son los impulsos de desplazamiento y una señal denominada «Nivel de escritura/lectura»

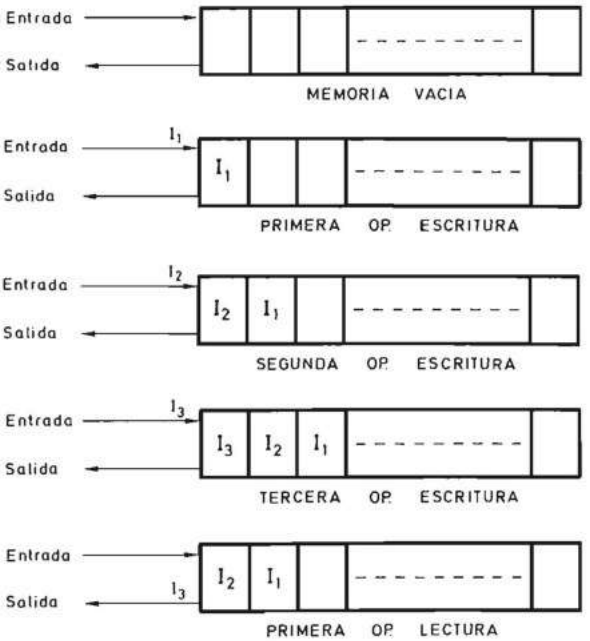


FIGURA 7.85.—**Forma de almacenamiento de la información en sucesivas operaciones de escritura y lectura de una memoria PILA (LIFO).**

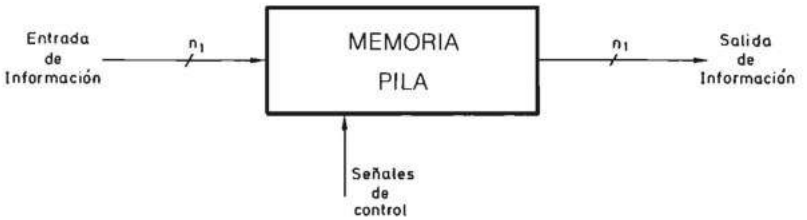


FIGURA 7.86.—**Diagrama de bloques básico de una memoria PILA (LIFO) organizada en posiciones de n_1 bits en paralelo.**

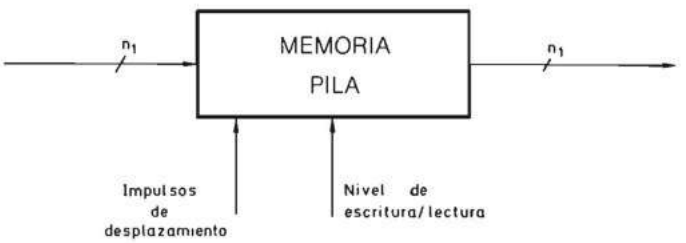


FIGURA 7.87.—**Diagrama de bloques de una memoria PILA (LIFO) en el que se indican explícitamente las señales de control.**

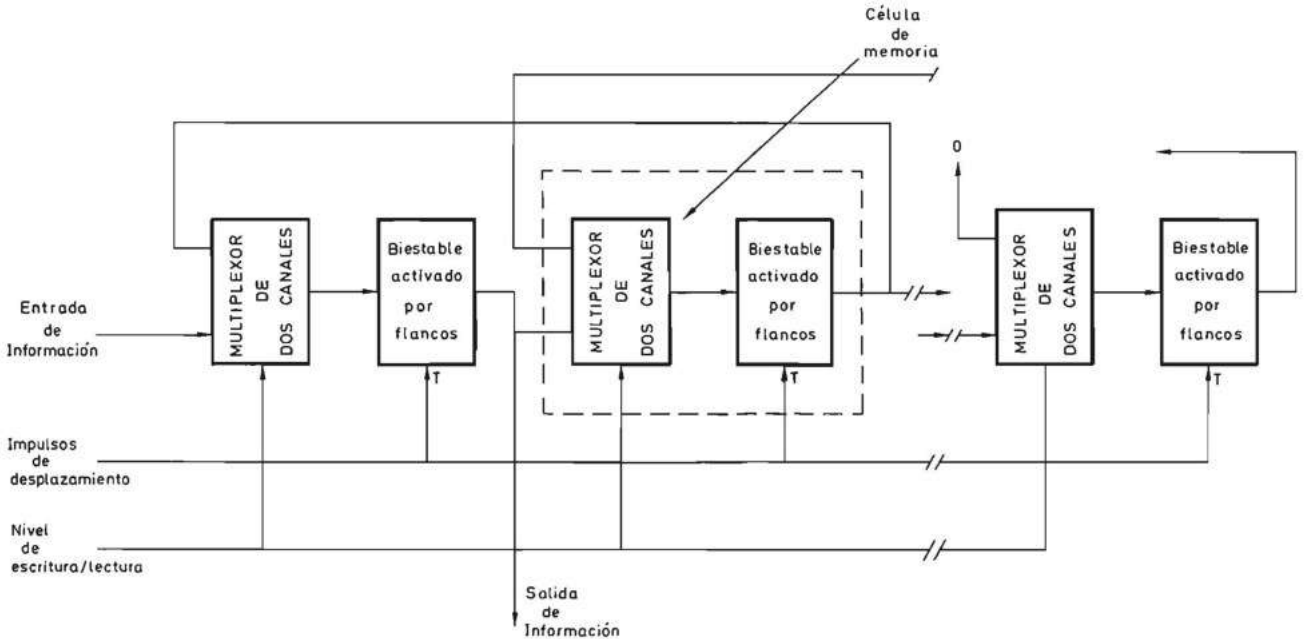


FIGURA 7.88.—**Memoria PILA realizada con un registro de desplazamiento estático reversible.**

cuyo estado lógico hace que los impulsos de desplazamiento trasladen la información hacia la derecha o hacia la izquierda (fig. 7.87).

La forma idónea de realizar una memoria PILA es mediante un registro de desplazamiento estático reversible tal como se indica en la figura 7.88. Un registro de desplazamiento estático reversible está compuesto de biestables síncronos y multiplexores de dos canales. Las entradas del multiplexor cuya salida está conectada a un determinado biestable, están unidas al biestable anterior y al siguiente al considerado. Por ejemplo, el multiplexor del primer biestable tiene una entrada unida a la salida del segundo biestable y la otra constituye la entrada externa de información a la memoria. La salida de la memoria está constituida por la salida del primer biestable. Las entradas T de todos los biestables se conectan entre sí y al terminal de impulsos de desplazamiento. Las variables de selección de todos los multiplexores se conectan entre sí y constituyen el «Nivel de escritura/lectura». El lector puede deducir sin dificultad que el sistema que acabamos de describir se comporta tal como se indicó para la memoria PILA. En la figura 7.88 se indicó solamente un único biestable. Repitiendo en paralelo n_1 veces la estructura de este esquema se tiene una memoria PILA organizada en serie posición a posición de n_1 bits cuyo diagrama de bloques es el de la figura 7.89 cuando está realizada con un registro de desplazamiento reversible. La entrada que decide el desplazamiento hacia la derecha o hacia la izquierda de la información en el interior del registro se denomina $\overline{D}/I$ lo cual indica que, si se encuentra en estado lógico cero al aplicar un impulso de desplazamiento, la información se desplaza hacia la derecha y si se encuentra en estado lógico uno ésta se desplaza hacia la izquierda. Desde un punto de vista funcional no tiene importancia la forma en que está realizada la memoria sino que lo importante son sus características de entrada y salida de información y, por ello, el diagrama de bloques de una memoria PILA es el indicado en la figura 7.90. Si se desea que los terminales de entrada y salida sean comunes, se les puede unir mediante un circuito seguidor con salida de tres estados,

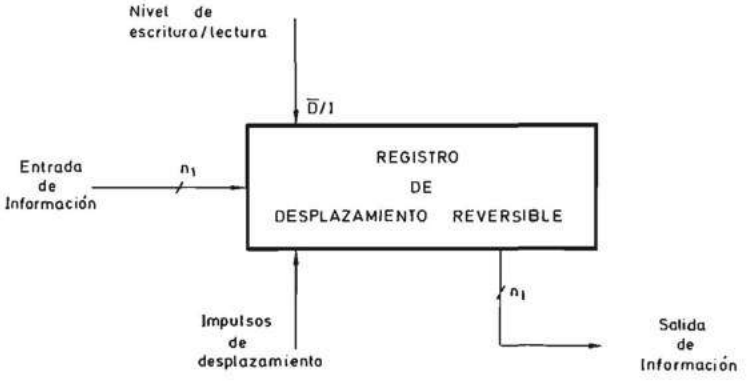


FIGURA 7.89.—**Diagrama de bloques de una memoria PILA de n_1 bits en paralelo, realizada con un registro de desplazamiento estático reversible.**

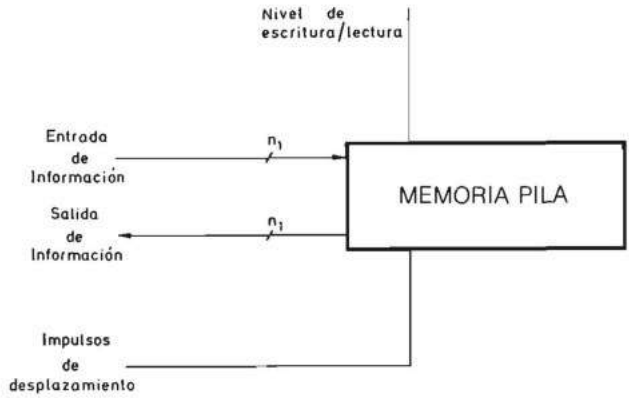


FIGURA 7.90.—**Diagrama de bloques de una memoria pila de n_1 bits en paralelo.**

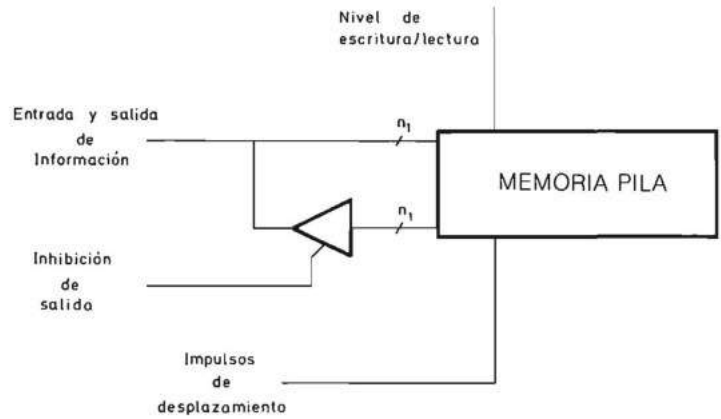


FIGURA 7.91.—**Memoria PILA con terminales únicos de entrada/salida y salida de tres estados.**

controlado por una señal de inhibición de salida tal como se indica en la figura 7.91. Se obtiene así el símbolo lógico representado en la figura 7.92.

7.2.3.3 Memorias asociativas (Content addressable memories) (CAM). Estas memorias se caracterizan porque la búsqueda de información en la operación de lectura no se realiza indicando una dirección y observando su contenido sino que se suministra la información a la memoria para observar si ésta la contiene en alguna de sus posiciones. La información almacenada en cada posición de una memoria asociativa puede estar organizada de dos formas fundamentales:

a) Dividida en dos partes o campos. Un campo es el de la información que se compara con la que se presenta a las entradas de la memoria en una operación de

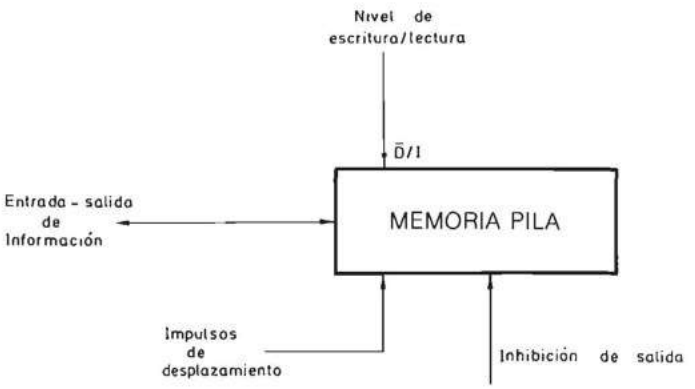


FIGURA 7.92.—**Símbolo lógico de una memoria PILA con terminales únicos de entrada/salida y salida de tres estados.**

lectura. El otro campo es el que da como respuesta la memoria cuando existe coincidencia entre el campo de comparación y la información de entrada.

b) Toda la información contenida en cada posición de la memoria se compara con la información de entrada. La memoria da solamente como respuesta, en general, un bit cuyo estado indica si existe alguna información en el interior de la memoria que es idéntica a la presentada a la entrada.

La estructura asociativa de una unidad de memoria es adecuada para el almacenamiento de ciertos tipos de organizaciones de los datos. Por ejemplo, una tabla de clientes con su dirección como dato asociado se debe organizar en forma asociativa para que al presentar a la memoria el nombre de un cliente, ésta dé a la salida su dirección.

La escritura de una memoria asociativa se puede realizar principalmente de dos formas.

a) Aleatoriamente.

En la operación de escritura se direcciona una posición de forma análoga a las memorias de acceso aleatorio y se introduce en ella la información externa.

b) Con lectura previa.

Previamente a la escritura, se realiza una lectura para saber si existe alguna posición de memoria que contenga una información idéntica a la que se quiere introducir en la memoria.

Si la respuesta es afirmativa, se reemplaza la información asociada al campo de comparación por la correspondiente presente a la entrada. Esto equivale a una puesta al día de la tabla indicada como ejemplo. Si la respuesta es negativa, se in-

troduce toda la información de entrada (ambos campos) en la primera posición vacía de la memoria.

La realización física de una memoria asociativa puede efectuarse de varios modos, de los cuales los principales son:

a) Con un comparador por cada posición de memoria. Esta es la organización asociativa propiamente dicha cuyo diagrama de bloques se indica en la figura 7.93. La lectura de esta memoria se hace de forma asociativa y la escritura se puede realizar de forma aleatoria o asociativa.

Cada posición de la memoria posee dos campos C_1 y C_2 de n_1 y n_2 bits respectivamente. Se forman dos barras de información, una de entrada de $n_1 + n_2$ bits y otra de salida de n_2 bits unidas a todas las posiciones. Estas reciben también la señal de «Control de escritura-lectura» que permite decidir la operación que se va a efectuar. La selección de cada posición se hace mediante una puerta O que recibe la salida de un decodificador y la del comparador que compara los n_1 bits de la memoria con los de la información externa de entrada. El decodificador posee dos entradas de inhibición que han de estar en estado uno simultáneamente para que se realice dicha acción. Una de estas señales es la de «Control de escritura-lectura» $C_{E/L}$ que sólo permite la desinhibición cuando se encuentra en el estado uno correspondiente a escritura. La otra es una señal de «Inhibición externa» I_E que actúa además sobre el comparador; cuando se encuentra en estado uno desinhibe al decodificador y cuando se encuentra en estado cero realiza la acción contraria. En la tabla 7.4 se indican las acciones que se realizan para cada combinación de I_E y $C_{E/L}$.

Analizaremos seguidamente el funcionamiento del circuito representado en la figura 7.93. Cuando las señales de «Inhibición externa» I_E y de «Control de escritura-lectura $C_{E/L}$ se encuentran ambas en estado cero lógico, el decodificador está inhibido y las posiciones de la memoria están preparadas para realizar la operación de lectura. La información de entrada del campo C_1 de n_1 bits se compara con la correspondiente a cada posición. Aquella posición cuyo contenido coincide con el de entrada, queda direccionada y la información de su campo C_2 de n_2 bits aparece en la barra de salida. Si la señal I_E continúa en estado lógico cero y la señal $C_{E/L}$ en estado uno, el decodificador está inhibido y el comparador está desinhibido y las posiciones de memoria están preparadas para realizar una operación de escritura. Al igual que en el caso anterior, la información del campo C_1 de entrada se compara con la correspondiente de cada posición y queda direcciona-

I_E	$C_{E/L}$	*Operación*
0	0	Lectura asociativa
0	1	Escritura asociativa
1	0	Inhibición
1	1	Escritura aleatoria

TABLA 7.4.— **Tabla de modo de funcionamiento de la memoria asociativa de la figura 7.93**

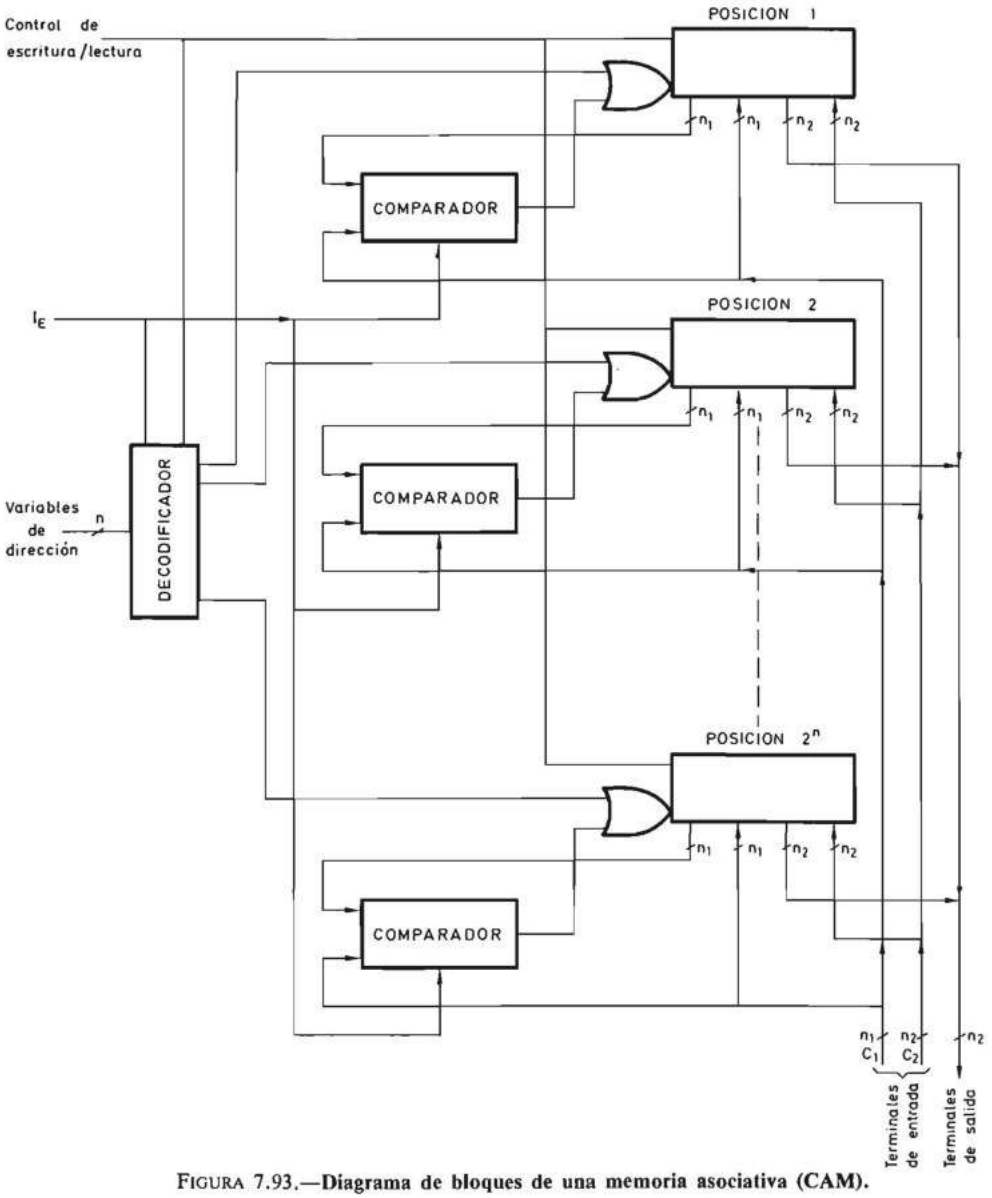


FIGURA 7.93.—**Diagrama de bloques de una memoria asociativa (CAM).**

da la posición en que coinciden. Pero en este caso, en lugar de realizarse una lectura, la información del campo C_2 de entrada se introduce en la célula correspondiente.

La escritura aleatoria se realiza desinhibiendo el decodificador e inhibiendo el comparador, para lo cual I_E y $C_{E/L}$ se han de encontrar ambas en estado uno. El sistema exterior ha de colocar en los terminales de entrada la información de am-

bos campos C_1 y C_2 y además la combinación de las variables de dirección correspondiente a la posición en la que se desea escribir. Naturalmente, el sistema externo ha de conocer las posiciones vacías para no destruir inadvertidamente la información contenida en una posición ocupada.

Por ser muy complicada esta realización física solamente se han integrado memorias de este tipo de pequeña capacidad.

b) Con una memoria de acceso aleatorio (RAM) y un único comparador. Mediante una unidad de control adecuadamente realizada, que puede ser cableada como las estudiadas en el capítulo 6, o programable, se puede realizar una lectura y escritura asociativas de la memoria de acceso aleatorio. Al recibir la unidad de control una orden de lectura, realiza un barrido secuencial de las posiciones de la memoria de acceso aleatorio a partir de la cero. En cuanto la información de entrada coincide con los n_1 bits del campo C_1 de entrada, la salida del comparador cambia de estado y hace que la unidad de control detenga el barrido; en los n_2 bits de salida se tiene entonces la información asociada (fig. 7.94). La escritura puede realizarse de forma asociativa o aleatoriamente. Para ello se conecta a la unidad de control una orden de escritura O_E y una señal de control de la forma de realizarse aquella C_E. Si se desea realizar la escritura asociativa, la unidad de control inicia un ciclo de lectura y cuando se verifica la coincidencia efectúa la escritura en la posición correspondiente. Cuando la escritura es aleatoria, se hace llegar a la memoria la dirección externa y se realiza la escritura en ella.

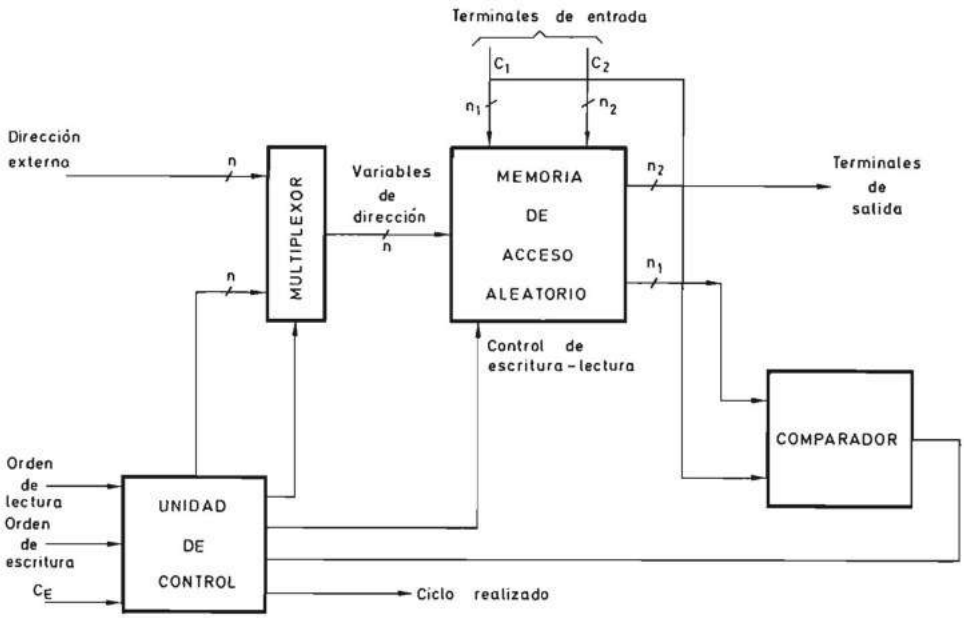


FIGURA 7.94.—**Memoria asociativa realizada con una memoria de acceso aleatorio activa y un sistema secuencial síncrono.**

Para informar al sistema externo de que la orden de escritura o lectura ha sido realizada, la unidad de control puede suministrar una señal de «Ciclo realizado».

7.3 TECNOLOGIAS DE LAS UNIDADES DE MEMORIA

7.3.1 Introducción

En los apartados anteriores se han estudiado las diferentes estructuras de las memorias desde un punto de vista especialmente funcional, sin tener en cuenta la forma de realizarlas físicamente, a fin de que el lector adquiriese un conocimiento claro de la forma en que se introduce (escribe) y se extrae (lee) la información en los diferentes tipos. Pero la experiencia docente del autor le ha demostrado que el funcionamiento de una estructura se comprende y memoriza mejor si se analiza la forma en que se realiza físicamente. El enorme esfuerzo investigador que se ha realizado en los últimos años y que se está realizando actualmente en el desarrollo de las tecnologías electrónicas hace que éstas evolucionen muy rápidamente y se queden anticuadas en poco tiempo. Por todo ello en los apartados siguientes se estudian algunas de las tecnologías más utilizadas en la realización de unidades de memoria para ayudar al lector a comprender mejor los conceptos teóricos estudiados en los párrafos anteriores y poder plasmarlos en sistemas físicos que actúen de acuerdo con aquéllos, sin pretender agotar el tema. Para un estudio más detallado de algún circuito en particular se remite al lector a las notas de aplicación de los fabricantes, algunas de las cuales se citan en la bibliografía.

7.3.2 Tecnologías de las memorias de acceso aleatorio (RAM)

Tal como se vio en el apartado 7.2.3.1.3, las memorias de acceso aleatorio se pueden clasificar de acuerdo con el parámetro permanencia en memorias activas o de escritura y lectura, y memorias pasivas. La realización física de ambos tipos difiere notablemente y por ello se estudian por separado.

7.3.2.1 Memorias activas. De acuerdo con lo expuesto en el apartado 7.2.3.1.3.1, las memorias de acceso aleatorio activas son aquellas que poseen unos tiempos de escritura y lectura del mismo orden de magnitud.

En la tabla 7.5 se clasifican las memorias activas de acuerdo con la tecnología en que están realizadas. Esta tabla es un complemento de la 7.1.

Existen dos elementos básicos utilizados en la realización de estas memorias:

a) los núcleos magnéticos de ferrita que memorizan la información debido a la elevada remanencia de su ciclo de histéresis.

b) Los elementos semiconductores en sus dos versiones: transistores bipolares y transistores MOS.

Antes de que las técnicas de integración alcanzasen su desarrollo actual, las memorias de ferrita eran las más utilizadas para la realización de memorias activas.

Tecnologías de las memorias de acceso aleatorio activas				Volatilidad
	Memorias de ferrita			No volátiles
	Memorias de semiconductores	Bipolares		Volátiles
		MOS	Estáticas	
			Dinámicas	

TABLA 7.5.— **Clasificación de las memorias de acceso aleatorio activas de acuerdo con la tecnología de realización.**

La principal característica de las memorias de ferrita es, tal como se indica en la tabla 7.5, la carencia de volatilidad, es decir, conservan la información en ausencia de la tensión de alimentación.

El gran avance de las tecnologías electrónicas ha permitido realizar en circuito integrado de gran escala de integración bloques funcionales de memoria de elevada capacidad y tiempo de acceso y ciclo muy reducidos (menores de 100 ns), no alcanzables con núcleos de ferrita. La desventaja de su volatilidad viene compensada en algunas tecnologías como, por ejemplo, la de MOS complementario (CMOS) por una reducida potencia necesaria para mantener la información estáticamente sin realizar operaciones de escritura o lectura. Esto permite la utilización de baterías recargables de reducido tamaño que mantienen la información cuando se produce un fallo de la red de alimentación.

Por todo lo expuesto se estudian solamente las memorias de acceso aleatorio activas realizadas con semiconductores.

Las células básicas bipolares capaces de memorizar un bit de información están basadas en la interconexión de dos inversores. En la figura 7.95 se indica el esquema de la de uso más extendido en la tecnología TTL estudiada en el apartado 5.4.4.3.3 Los transistores T_1 y T_2 realimentados poseen varios emisores. En la figura 7.95 se supone que la célula forma parte de una memoria con selección lineal (2D) (estudiada en el apartado 7.2.3.1.2 y por ello sólo posee una entrada de selección unida a un emisor de cada transistor. Si la célula se seleccionase por coincidencia (3D), poseería dos líneas de selección conectadas a sendos emisores de ambos transistores. Cuando la célula no está seleccionada, la línea de selección se encuentra al nivel de masa. En esta situación uno de los dos transistores T_1 o T_2 conduce y la célula memoriza un cero o un uno. Para leer el contenido de una célula, es decir, ver su contenido, se eleva al valor de V_{cc} la tensión de su línea de selección y esto hace que la corriente que pasa por el emisor de T_1 o de T_2 unida a la línea de selección se desvíe al emisor unido a la salida Q o $\overline{Q}$ respectivamente y a la salida de los amplificadores de lectura L_1 y L_2 se observe el estado de la célula.

Para memorizar el estado de las entradas I e $\overline{I}$ en el interior del biestable se realiza la selección tal como se indicó en el apartado anterior y simultáneamente

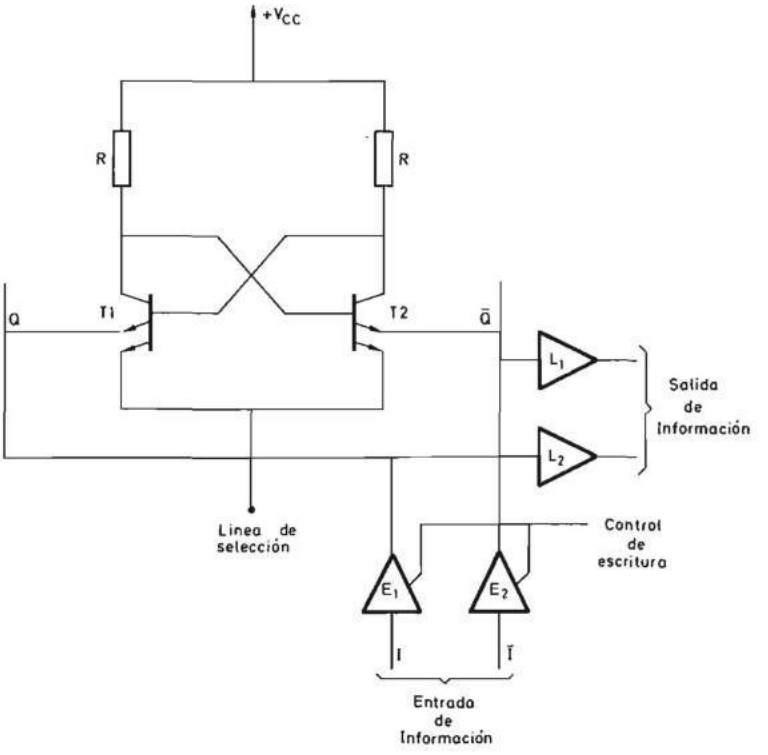


FIGURA 7.95.—**Célula de una memoria de acceso aleatorio estática bipolar realizada en tecnología TTL.**

se aplica la señal de control de escritura que desinhibe los amplificadores de escritura E_1 y E_2. La salida de uno sólo de estos amplificadores se pondrá al nivel de masa y hará conducir al transistor T_1 o T_2 correspondiente. Al volver la línea de selección al nivel de masa, dejará de conducir el transistor T_1 o T_2 a través de su emisor unido a Q o $\overline{Q}$ y lo ha hará a través de aquélla, memorizándose de esta forma la información I.

En la figura 7.95 se supone que la información que se escribe en el biestable se lee simultáneamente a la salida de los amplificadores L_1 y L_2. Si se quiere inhibir la lectura mientras se realiza la escritura, se inhibirán dichos amplificadores mediante la señal de control de escritura.

Las memorias activas de tecnología MOS presentan la ventaja de las dimensiones reducidas de su célula básica, que carece de resistencias porque, tal como se estudió en el apartado 5.4.4.2, la carga de los transistores MOS es a su vez otro transistor MOS.

Las memorias activas MOS se pueden clasificar en dos tipos principales que se diferencian por el principio físico utilizado por la célula básica para memorizar la información.

Las memorias estáticas son aquéllas que memorizan la información mediante dos inversores interconectados formando un biestable activado por niveles. Las dos tecnologías más utilizadas para realizar estas células son la de MOS canal *N* y la de MOS complementario (CMOS). En la figura 7.96 se representa la célula básica realizada con transistores MOS canal *N* enriquecidos y el lector puede observar su paralelismo con la célula bipolar de la figura 7.95. Los transistores T_3 y T_4 actúan como resistencia de carga de T_1 y T_2 respectivamente. La salida directa Q de la célula es el punto común al surtidor de T_3 y el drenador de T_1 y la salida $\overline{Q}$ es el mismo punto de T_4 y T_2. Veremos seguidamente cómo se puede introducir información en esta célula o leer la que contiene. En primer lugar es necesario dotarla de los dispositivos necesarios para seleccionarla, que se indican en la figura 7.97 y que consisten en los transistores T_5 y T_6 cuyas puertas están unidas entre sí y constituyen la línea de selección.

Para leer el contenido de la célula se aplica el nivel de tensión V_{DD} a la línea de selección y de esta forma se hacen conducir los transistores T_5 y T_6. El estado de las líneas I e $\overline{I}$ se hace igual al de las salidas de la célula Q y $\overline{Q}$ y puede ser aplicado a un amplificador cuya salida se conecta al exterior.

Para escribir en la célula una vez seleccionada, se hace llegar la información externa a las líneas I e $\overline{I}$.

Para que el lector comprenda mejor los conceptos que acabamos de explicar, representaremos la célula de la figura 7.97*a* mediante el diagrama de bloques de la figura 7.97*b* y realizaremos con él una memoria de selección lineal (2D) y otra de selección por coincidencia (3D).

En la figura 7.98 se representa una memoria de selección lineal de 16 posiciones

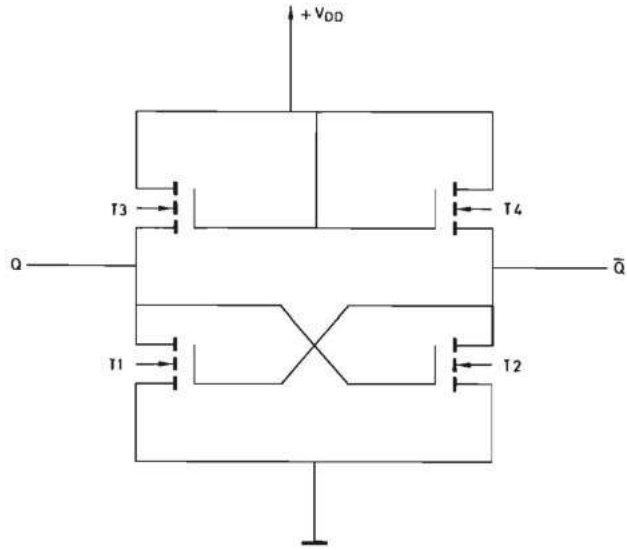


FIGURA 7.96.—**Célula de una memoria de acceso aleatorio estática realizada con transistores MOS de canal N.**

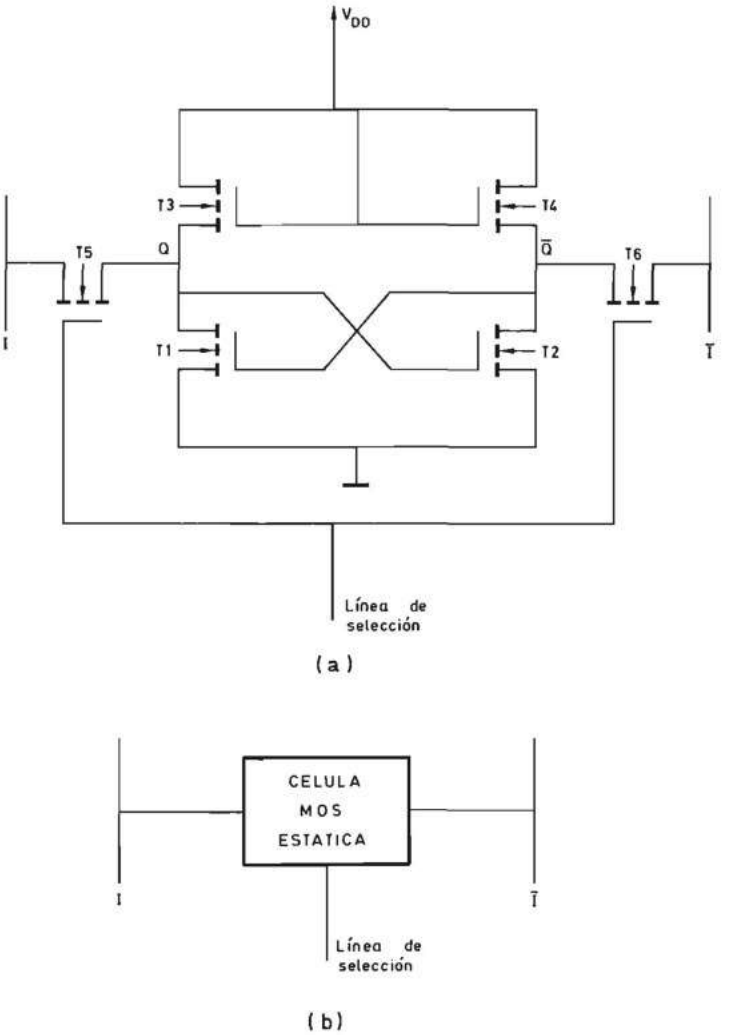


FIGURA 7.97.—**Célula de una memoria de acceso aleatorio estática, realizada con transistores MOS de canal N, con dispositivos de selección: *a*) Esquema del circuito; *b*) Diagrama de bloques.**

de un bit cada una realizada con las células MOS estáticas de la figura 7.97. Para seleccionar las 16 posiciones se necesitan cuatro variables de dirección ($2^4 = 16$) que se conectan a un decodificador hexadecimal (uno entre dieciséis). Cada una de las 16 salidas de este decodificador se conecta a la línea de selección de una célula. Al aplicar una combinación de las variables de dirección a la entrada del decodificador, se selecciona la célula cuya entrada de selección está unida a la salida del decodificador que está activada. Las salidas I e $\bar{I}$ de todas las células están unidas entre sí formando una barra de información y en ella aparece el contenido de la célula seleccionada.

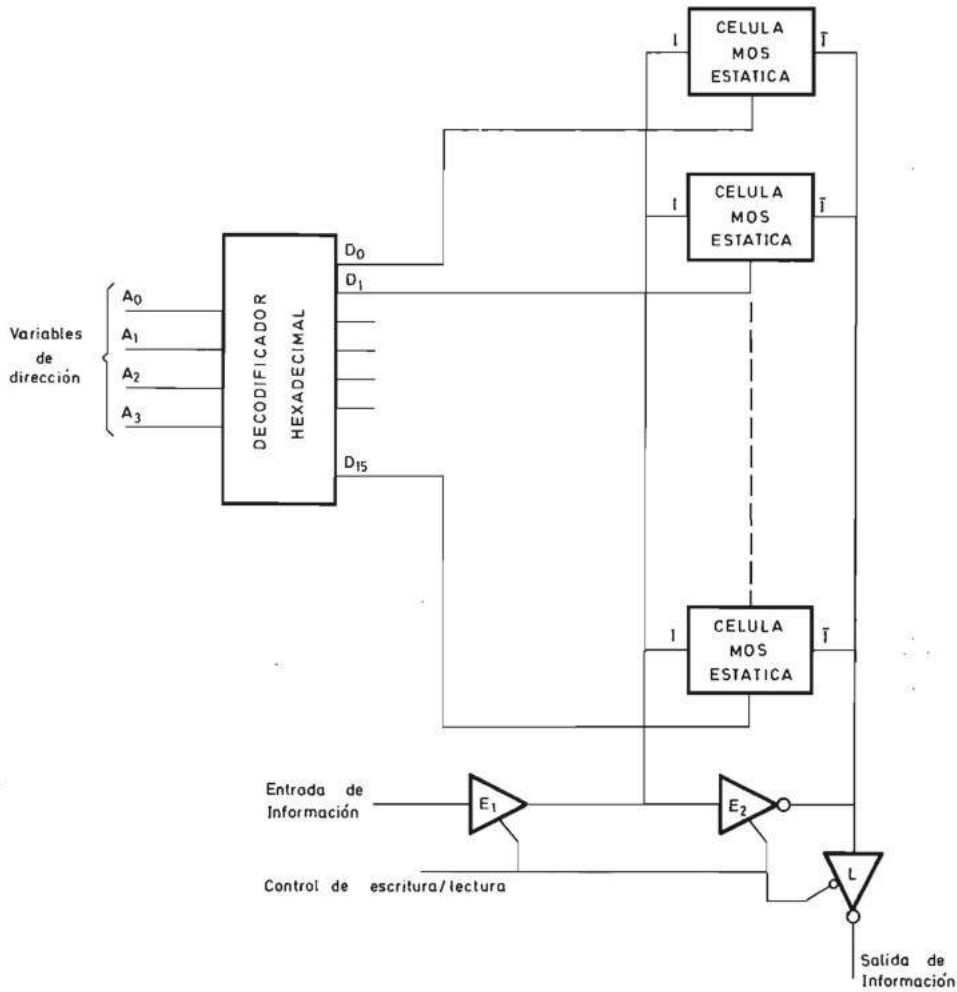


FIGURA 7.98.—**Memoria estática de 16 × 1 organizada 2D y realizada con la célula básica de la figura 7.97.**

Si la señal de «Control de escritura-lectura» se encuentra en el nivel correspondiente a la lectura, los amplificadores E_1 y E_2 están inhibidos y el estado de la barra $\bar{I}$ aparece en la salida a través del amplificador inversor de lectura L. Si, por el contrario, la señal de «Control de escritura-lectura» se encuentra en el nivel correspondiente a escritura, la información de entrada aparece a través de E_1 en la barra I y su inverso en la barra $\bar{I}$; de esta forma la célula memoriza el estado de la entrada.

En la figura 7.98 que acabamos de analizar se representa el esquema de una memoria estática MOS de selección lineal (2D) de 16 posiciones de un bit cada una. El conjunto de las células y los amplificadores de escritura y lectura constitu-

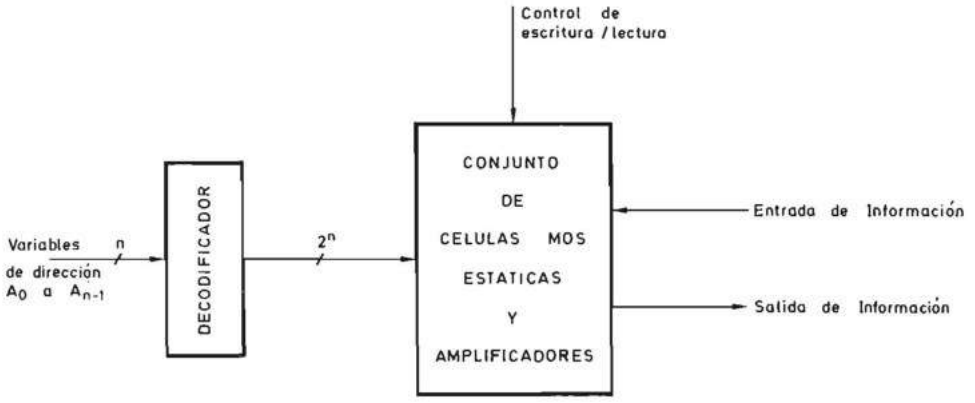


FIGURA 7.99.—**Diagrama de bloques de una memoria de acceso aleatorio estática de 2^n posiciones de un bit realizada con la estructura de la figura 7.98.**

yen un bloque funcional que puede representarse mediante un rectángulo con los terminales de selección, entrada y salida de información y «Control de escritura-lectura» tal como se indica en la figura 7.99.

La realización de una memoria de más de un bit en cada posición se comprende fácilmente a partir del diagrama de bloques de la figura 7.99.

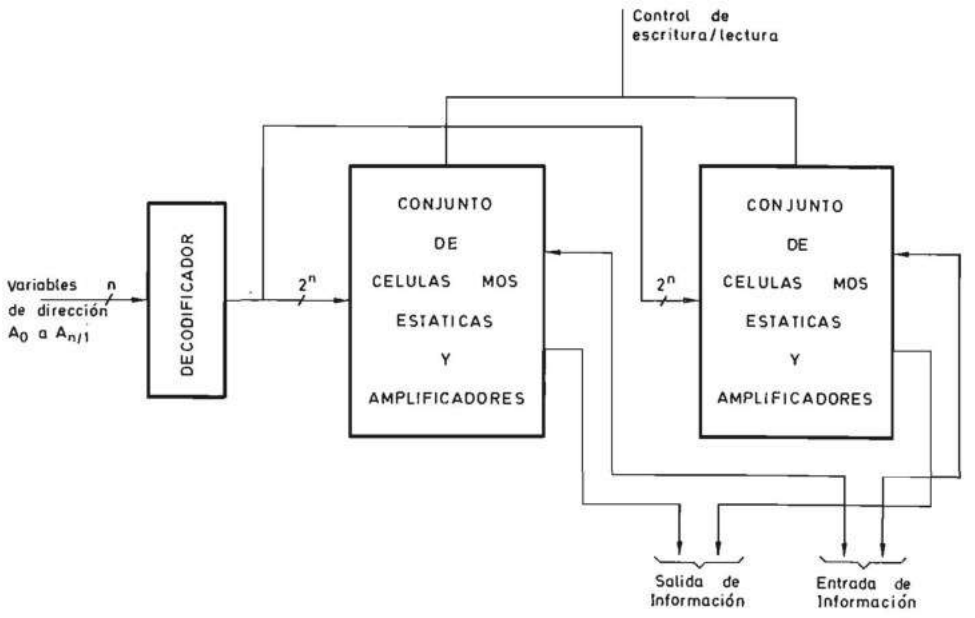


FIGURA 7.100.—**Diagrama de bloques de una memoria de acceso aleatorio estática de 2^n posiciones de dos bits cada una con organización 2D.**

Una memoria de n_1 bits en cada posición se realiza mediante n_1 conjuntos de células MOS estáticas y amplificadores de escritura y lectura, con sus entradas y salidas de información en paralelo. Cada salida del decodificador se conecta a una célula de cada conjunto y la señal de «Control de escritura-lectura» es común a todos ellos. En la figura 7.100 se representa el esquema de bloques de una memoria de 2^n posiciones de dos bits en cada posición.

Una memoria de selección por coincidencia utiliza las mismas células estáticas representadas en la figura 7.97. En la figura 7.101 se representa el esquema de una

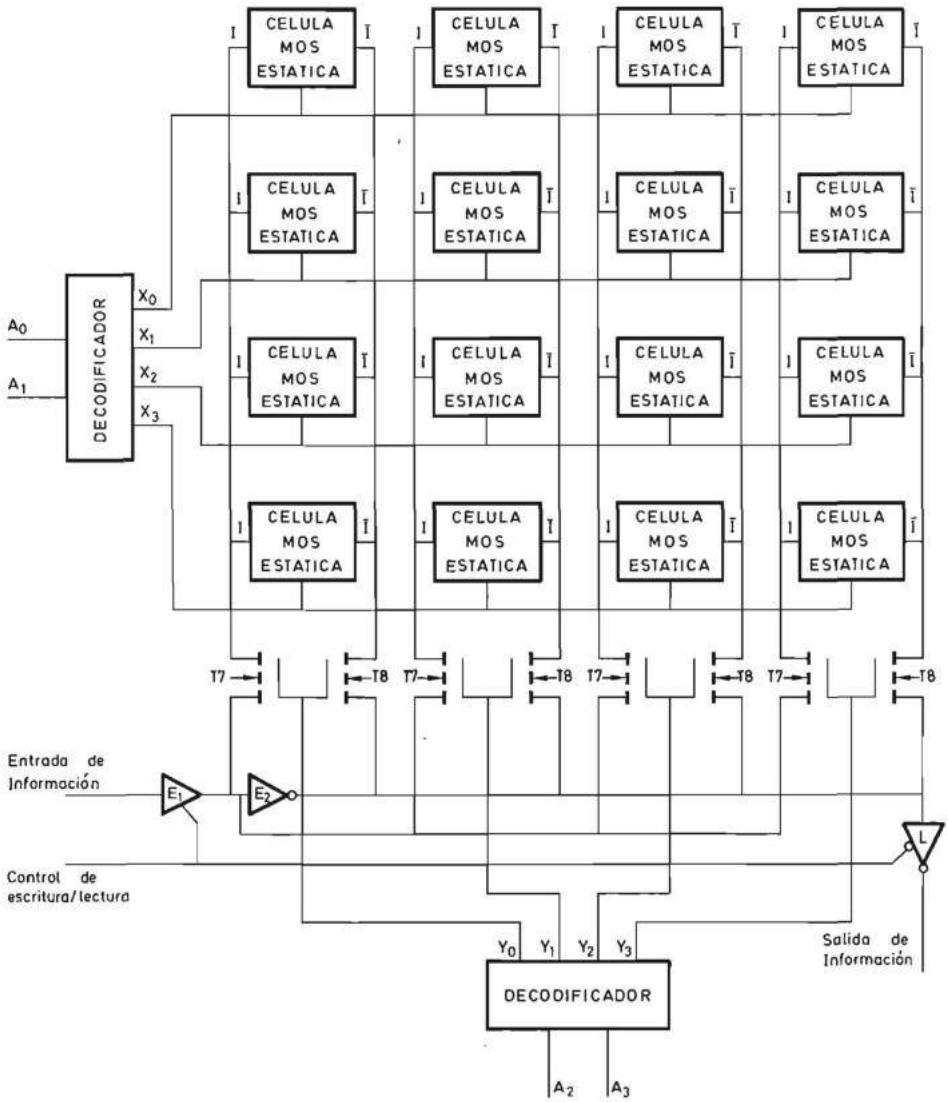


FIGURA 7.101.—**Memoria de acceso aleatorio estática de 16 × 1 con organización 3D.**

memoria de 16 posiciones de 1 bit cada una seleccionadas por coincidencia. Las cuatro variables de dirección se decodifican mediante dos decodificadores. Las células se disponen en filas y columnas. Las entradas de selección de todas las células de la misma línea se unen entre sí y se conectan a una salida x_i del decodificador que decodifica las variables A_0 y A_1. Las salidas I e $\bar{I}$ de todas las células de la misma columna se conectan entre sí y a los amplificadores de escritura y lectura a través de dos interruptores T_7 y T_8 constituidos por dos transistores MOS. Las puertas de estos transistores se conectan a las salidas del decodificador que decodifica las variables de dirección A_2 y A_3. La salida de este decodificador correspondiente a la combinación de A_2 y A_3 presente a la entrada, se activa y hace conducir a los transistores T_7 y T_8 unidos a ella. En los surtidores de ambos transistores que están unidos a las salidas de los amplificadores E_1 y E_2 aparece la información de la célula de la columna correspondiente cuya línea de selección está activada mediante una salida del decodificador de filas.

El conjunto de las células estáticas MOS, los transistores T_7 y T_8 y los amplificadores de escritura y lectura se puede considerar un bloque funcional representado en la figura 7.102 mediante un cuadrado. Una memoria de n bits en cada posición se realiza de una forma similar a la de selección lineal. En la figura 7.103 se representa el esquema de una memoria de 2^n posiciones de dos bits cada una,

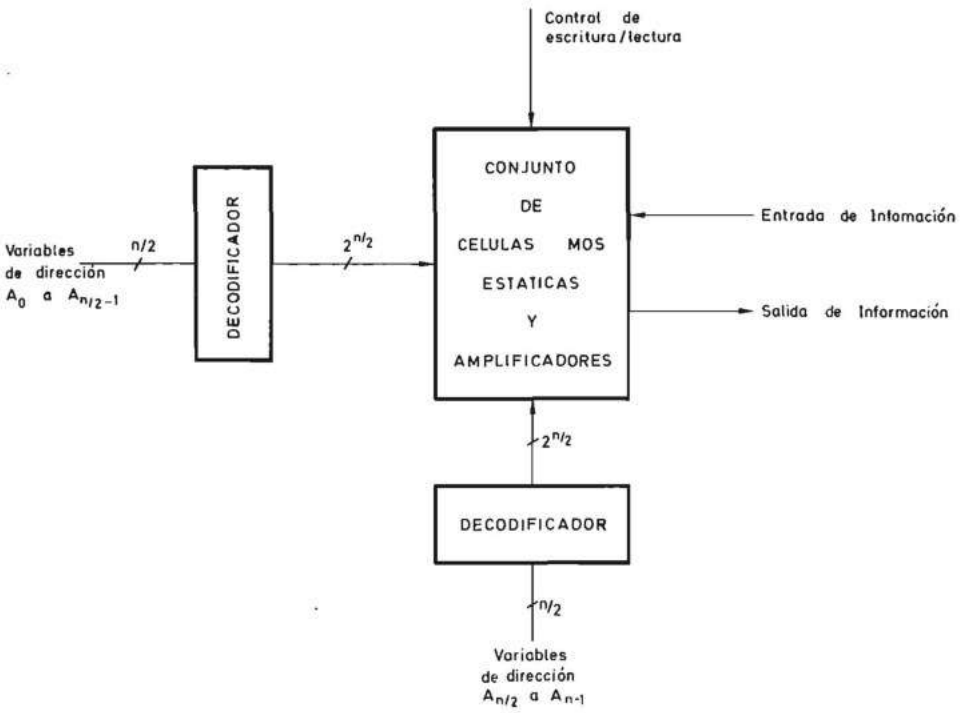


FIGURA 7.102.—**Diagrama de bloques de una memoria de acceso aleatorio estática de 2^n posiciones de un bit realizada con la estructura de la figura 7.101.**

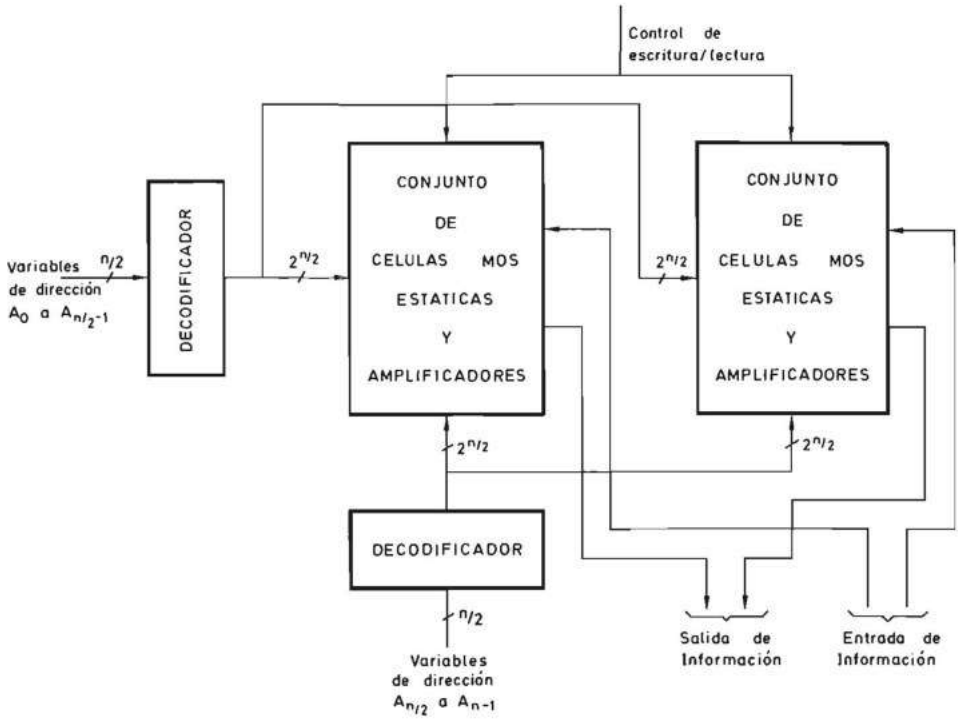


FIGURA 7.103.—**Diagrama de bloques de una memoria de acceso aleatorio estática de 2^n posiciones de dos bits cada una, con organización 3D.**

formada por dos conjuntos de células seleccionadas en paralelo por los mismos decodificadores. La señal de «Control de escritura-lectura» es común a ambas y la entrada y salida de cada conjunto se disponen en paralelo al exterior.

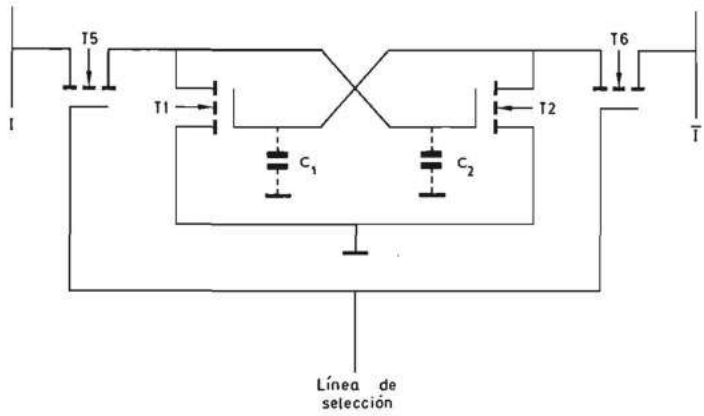


FIGURA 7.104.—**Célula básica de una memoria de acceso aleatorio dinámica con dos transistores MOS.**

Las células estáticas presentan la propiedad de almacenar la información indefinidamente mientras se mantiene la tensión de alimentación, pero constan de cuatro transistores (fig. 7.97) y, por lo tanto necesitan un área de semiconductor elevada. Para reducir el tamaño de la célula y elevar la capacidad de las unidades de memoria disponibles en un solo circuito integrado se crearon las memorias dinámicas. La impedancia entre la puerta y el surtidor de un transistor MOS es casi totalmente capacitiva y puede ser utilizada para memorizar una información por un tiempo finito del orden de algunos milisegundos. De esta forma se pueden suprimir los transistores T_3 y T_4 de la célula de la figura 7.97, obteniéndose la de la figura 7.104. Los condensadores C_1 y C_2 están constituidos por la capacidad parásita entre puerta y surtidor de los respectivos transistores.

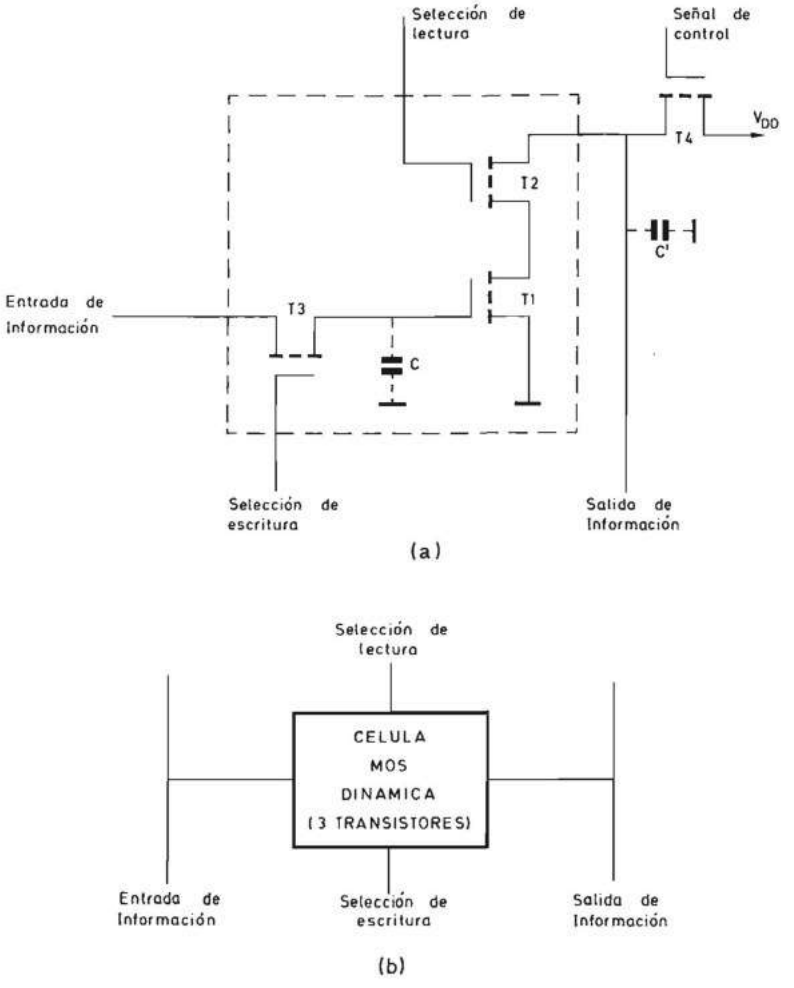


FIGURA 7.105.—**Célula básica de una memoria de acceso aleatorio dinámica con tres transistores** *a*) **Esquema del circuito;** *b*) **Diagrama de bloques.**

Tratando de reducir el número de transistores por célula se diseñó la representada en la figura 7.105 que está constituida por tan sólo tres transistores y tiene líneas de selección independientes para la lectura y la escritura. La información se almacena en la capacidad parásita existente entre la puerta y el surtidor del transistor T_1. Para introducir información en la célula se activa la línea de selección de escritura con lo cual se hace conducir a T_3 y el estado de la entrada de información se almacena en forma de carga en el condensador C. Para observar el estado de la célula se aplica en primer lugar la señal de control a la puerta de T_4 para que conduzca y que de esta forma el condensador C' se cargue a la tension V_{DD}. Seguidamente se activa la línea de selección de lectura con lo cual se satura T_2; si el condensador C está cargado, T_1 está saturado y la línea de salida de información se pone al nivel de masa. Si, por el contrario, C está descargado, la línea de salida de información se mantiene al nivel de V_{DD}. En la figura 7.105*b* se representa el diagrama de bloques de esta célula.

Ni la célula de la figura 7.104 ni la de la figura 7.105 almacenan la información por tiempo indefinido. Por esto, aunque no se realicen operaciones de escritura y lectura, es necesario leer periódicamente el estado de las células y volver a introducirlo en ellas en una operación que se denomina regrabado (refreshing).

Para ello se coloca un amplificador entre la salida y la entrada de información tal como se indica en la figura 7.106. Al aplicar la señal de control al amplificador

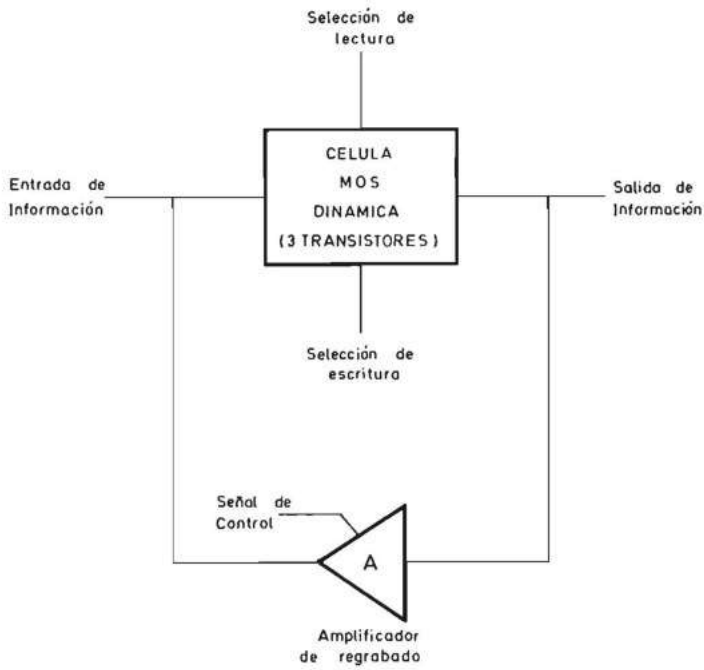


FIGURA 7.106.—**Célula de memoria dinámica con amplificador de regrabado (refreshing).**

A, éste presenta a la entrada la información contenida en la propia célula y se produce el regrabado. Las células dinámicas se suelen agrupar en una estructura de selección por coincidencia (3D) para formar una unidad de memoria y no es necesario un amplificador por célula, sino por columna, realizándose simultáneamente el regrabado de una fila y en secuencia el de las diferentes filas.

La necesidad del ciclo de regrabado de las filas de una memoria dinámica introduce una complicación no necesaria en las estáticas, que viene compensada por la mayor sencillez de la célula.

Tal como se estudia en el apartado 7.2.3.1.3*a* las memorias dinámicas han de poseer una unidad de control que genere automáticamente ciclos de regrabado. Para su estudio se remite al lector al citado apartado.

Las memorias MOS dinámicas han evolucionado gracias al desarrollo de la microelectrónica hasta alcanzar la capacidad de 1 Mbit en un único circuito integrado. En los manuales de datos de los fabricantes puede hacerse un estudio detallado de las mismas [INTE 86], [NEC 87], [TEXA 89].

7.3.2.2 Memorias pasivas. Tal como se explicó en el apartado 7.2.3.1.3.2, las memorias de acceso aleatorio pasivas poseen tiempos de lectura y de escritura diferentes. Su principal característica es la ausencia de volatilidad lo cual las hace idóneas para el almacenamiento de las instrucciones de los sistemas digitales programables. Otra de las ventajas de las memorias pasivas es la reducida complejidad de su célula básica que contiene sólo uno o dos elementos semiconductores. Esta reducción de la complejidad trae consigo la disponibilidad en circuito integrado de bloques funcionales de mayor capacidad que los de las memorias activas y un tiempo de acceso mucho menor, lo cual es muy importante cuando la memoria pasiva contiene las instrucciones de un sistema digital programable y está conectada a una unidad central de proceso (CPU) a través de una barra de información.

En la tabla 7.6, que complementa a la 7.2, se clasifican las memorias pasivas de acuerdo con el tipo de dispositivo utilizado en su realización.

Las tecnologías bipolares permiten actualmente la realización de memorias totalmente pasivas (ROM) o pasivas programables (PROM). Las memorias total-

Memorias pasivas de semiconductores	Bipolares	Totalmente pasivas (ROM)
		Pasivas programables (PROM)
	MOS	Totalmente pasivas (ROM)
		Pasivas programables (PROM)
		Pasivas reprogramables (RPROM)

TABLA 7.6.— **Clasificación de las memorias pasivas de acuerdo con la tecnología de realización.**

mente pasivas las programa el fabricante. La forma de realizar la programación depende del proceso de fabricación. En la figura 7.107 se representa el esquema de una memoria totalmente pasiva (ROM) de 2^n posiciones de cuatro bits cada una de selección lineal (2D). La célula básica está formada por un transistor *NPN*. Los cuatro transistores que constituyen una posición de memoria tienen sus bases conectadas entre sí y unidas a una de las salidas del decodificador que recibe en sus entradas las n variables de dirección. Todos los transistores de las diferentes posiciones de la memoria que almacenan el bit de idéntico peso tienen sus emisores conectados entre sí y a la etapa de salida constituida por un transistor T montado en emisor común y las resistencias R_1 y R_2. Cuando una salida del decodificador está activada, aplica una tensión $+ V$ a la base de todos los transistores que están unidos a ella, tensión que hace pasar una corriente a través de su unión base-emisor, la resistencia R_1 y la unión base emisor de los transistores T de salida.

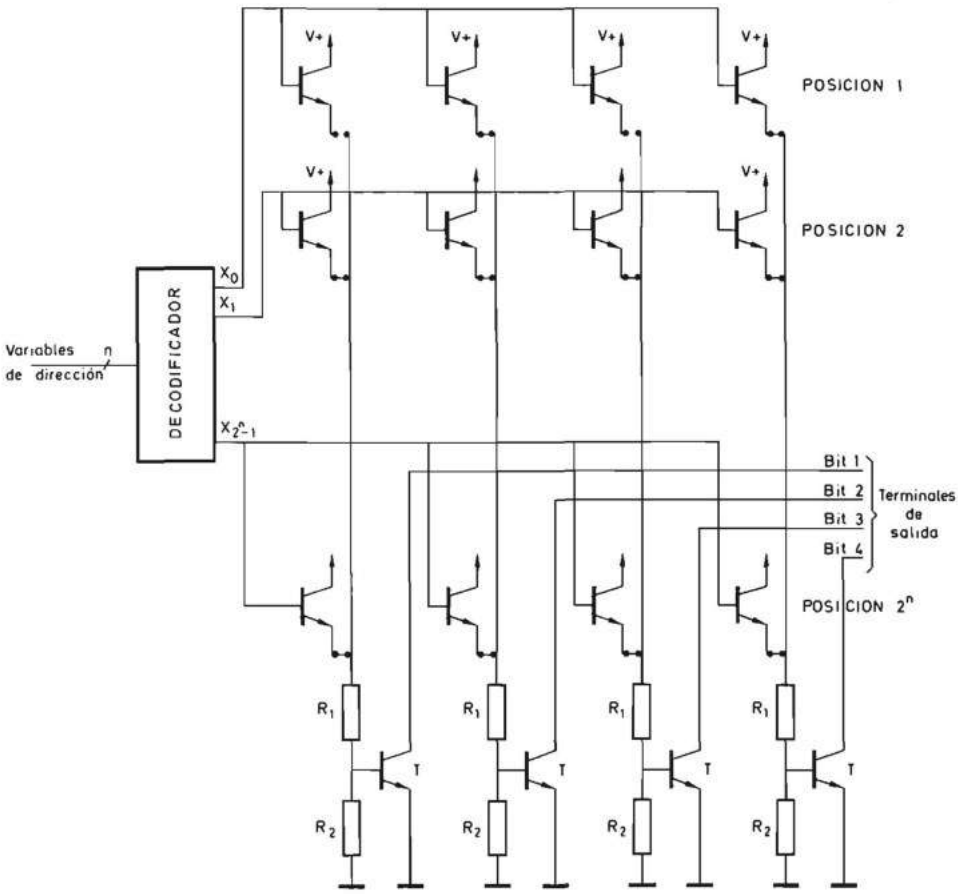


FIGURA 7.107.—**Esquema del circuito de una memoria totalmente pasiva bipolar realizada con organización 2D.**

De esta forma las salidas se ponen al nivel de masa (cero lógico en lógica positiva). La existencia de un transistor en todas las células de la memoria hace que ésta almacene un cero en todas ellas. El almacenamiento de un uno se realiza eliminando la conexión entre el emisor y la línea unida a la resistencia R_1. Por ejemplo, en la posición 1 de la memoria representada en la figura 7.107 se almacena de izquierda a derecha la información 1010 en lógica positiva.

Las memorias totalmente pasivas presentan el inconveniente de su falta de flexibilidad, lo cual no permite su utilización en aplicaciones en que el número de unidades es reducido. Por ello los fabricantes crearon las memorias pasivas programables (PROM) que se diferencian de las totalmente pasivas solamente en la existencia de un fusible de gran precisión, en serie con el elemento que constituye la célula de memoria (fig. 7.108). Inicialmente todos los fusibles están intactos y la programación la realiza el usuario haciendo pasar una corriente de magnitud controlada por los fusibles correspondientes a aquellas células en las que se desea grabar un uno (en el convenio de lógica positiva).

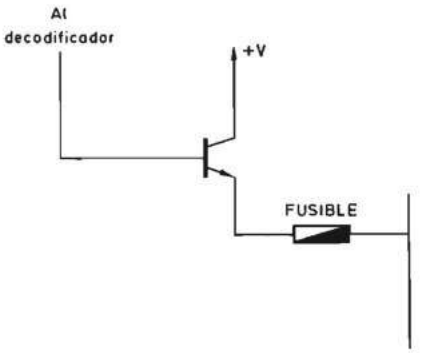


FIGURA 7.108.—**Célula básica de una memoria pasiva programable bipolar.**

Las memorias pasivas MOS tienen una configuración similar a las bipolares con la única diferencia del tipo de elemento semiconductor utilizado.

En la figura 7.109 se representa una memoria pasiva de selección por coincidencia (3D) realizada con transistores MOS canal N enriquecidos, de 2^n posiciones de un bit. Las n variables de dirección se decodifican mediante dos decodificadores: uno genera las variables de selección X de las filas y otro las Y de las columnas. Cada salida del decodificador de columnas va unida a la puerta de un transistor T_C cuyo drenador se une a la salida de información. Cada salida del decodificador de filas va unida a la puerta de $2^{n/2}$ transistores T_L, uno por columna. El surtidor de cada uno de estos transistores va unido a la tensión correspondiente al cero lógico (se utiliza el criterio de lógica positiva). Por lo tanto, existen tantos transistores T_L como posiciones de memoria; estos transistores constituyen el elemento de memoria. Cuando se aplica una dirección a las entradas

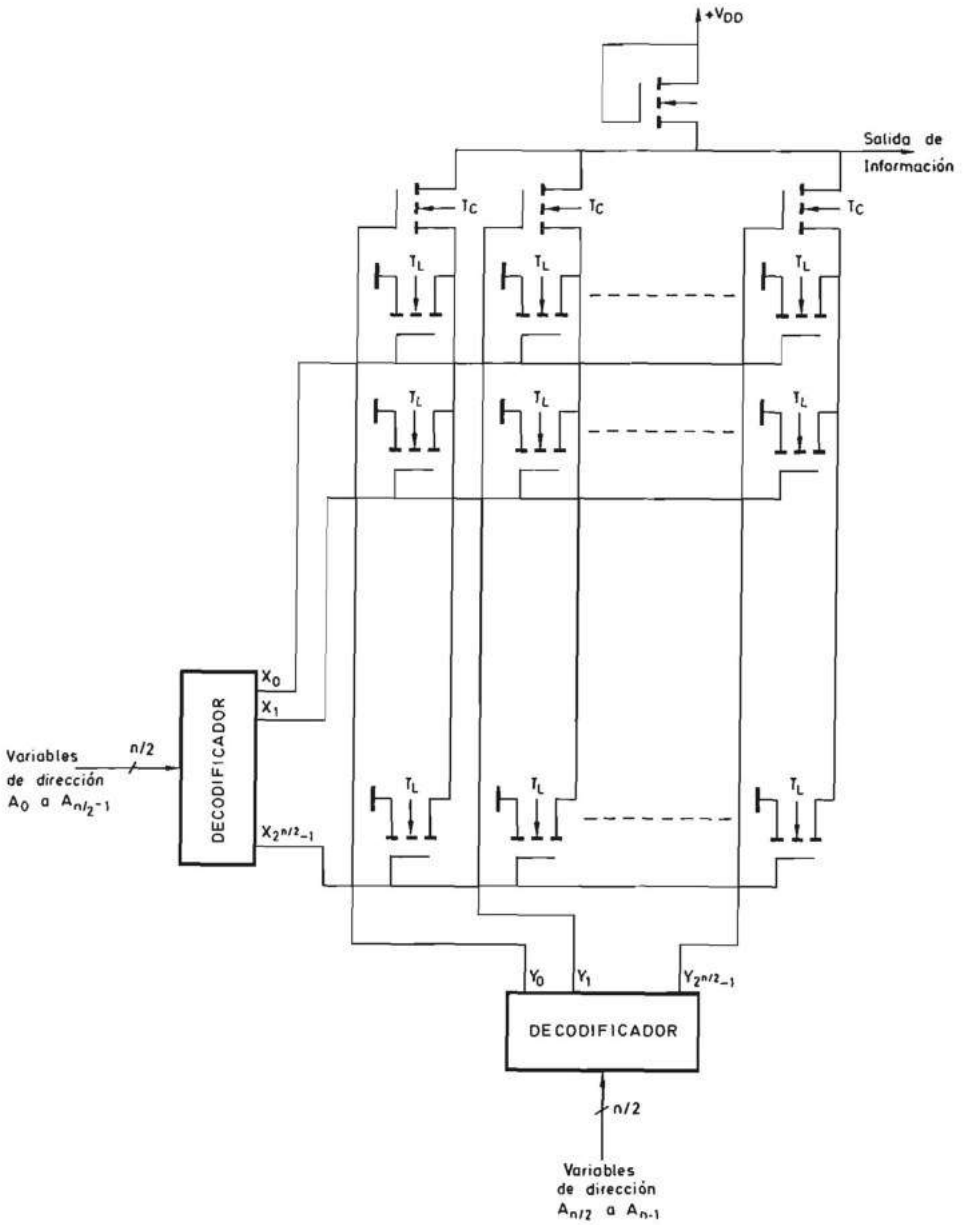


FIGURA 7.109.—**Memoria pasiva con transistores MOS canal N realizada con organización 3D.**

de ambos decodificadores se activa una salida de cada uno y, por lo tanto, se aplica una tensión de conducción a la puerta de un transistor T_C y de todos los transistores T_L de una fila. Sólo un transistor T_C y otro T_L unido a él conducen si-

multáneamente y de esta forma el terminal de salida se pone al nivel del cero lógico. Si en la unión de cada fila con cada columna existe un transistor T_L, en cada célula se almacena un cero lógico. La programación de la memoria consiste en suprimir aquellos transistores T_L correspondientes a las células en las que se desea almacenar un uno lógico. Existen tres formas diferentes de realizar esta programación que dan lugar a otros tantos tipos de memorias pasivas MOS:

a) En el proceso de fabricación del circuito, haciendo mayor de lo normal el espesor de la puerta de los transistores correspondientes a las células que deben almacenar un uno lógico. Esto hace que el transistor no conduzca aunque se aplique una tensión de activación a su puerta. Este procedimiento da lugar a las memorias MOS totalmente pasivas (ROM) que solamente son programables por el fabricante, pero no por el usuario.

b) Utilizando transistores especiales que poseen una puerta flotante que se representa en la figura 7.110*a*. La puerta externa se conecta a la línea de selección correspondiente del decodificador y la puerta flotante actúa como elemento de almacenamiento de cargas. La célula se programa poniendo a cero su substrato y aplicando una tensión elevada (típica de 21 V) que provoca una inyección por avalancha de electrones que quedan atrapados en la puerta flotante. En el estado inicial (sin electrones en la puerta), el transistor posee una tensión de umbral (V_{TH}) baja que permite su activación cuando se aplican 5 V a la puerta externa con res-

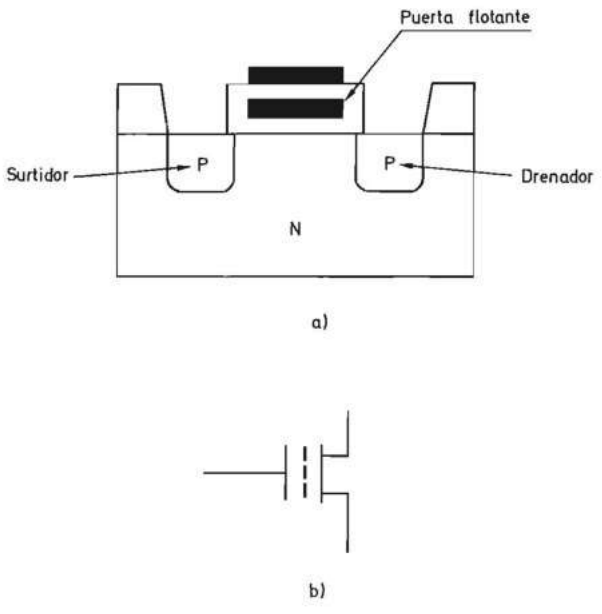


FIGURA 7.110.—**Transistor MOS canal N de puerta flotante: *a*) Realización física; *b*) Símbolo eléctrico.**

pecto al surtidor. La presencia de electrones en la puerta flotante eleva la tensión de umbral (V_{TH}) y hace que el transistor no se active aunque se apliquen 5 V a la puerta externa.

Una memoria formada por transistores como el descrito se programa seleccionando secuencialmente las posiciones e inyectando o no electrones en las correspondientes puertas aisladas. Si se quiere cambiar el contenido de una memoria ya programada es necesario previamente eliminar los electrones de todas las puertas flotantes que los contienen mediante la aplicación de rayos ultravioleta al conjunto de células. Las memorias formadas por estas células reciben el nombre de memorias pasivas programables eléctricamente EPROM (Erasable programmable read only memories) que hace referencia a su grabación mediante impulsos eléctricos. En el apartado 7.2.3.1.3.2 se estudian las memorias EPROM desde un punto de vista funcional y se indican sus aplicaciones.

c) Utilizando transistores de puerta flotante en los que se ha reducido, con respecto a los analizados en el apartado anterior, el espesor de la capa aislante que separa la puerta flotante de la puerta externa y del drenador. Se logra de esta manera que tanto la inyección de electrones como su eliminación de la puerta flotante pueda realizarse mediante impulsos eléctricos. Para lograr una unidad de memoria fiable con estos transistores es necesario que la célula básica esté formada por dos transistores: uno de puerta flotante y otro normal colocado en serie con el primero que tiene como misión seleccionar el transistor que se borra o graba mediante impulsos eléctricos.

Las memorias formadas por transistores como los descritos reciben el nombre de memorias pasivas borrables y programables eléctricamente EEPROM o E^2PROM (Electrically erasable programmable read only memories).

En la figura 7.111 se representa el esquema básico de una memoria EEPROM de 2^n posiciones de 1 bit cada una. Para tener n_1 bits es necesario disponer n_1 matrices como la indicada gobernadas todas por dos únicos decodificadores.

Cada célula de memoria de la figura 7.111 está formada por un transistor normal T_L y otro de puerta aislada T_p. El transistor T_p está en corte si su puerta aislada no posee portadores de carga y está saturado en caso contrario. La grabación consiste en inyectar dichos portadores mediante la aplicación de una tensión elevada a la puerta con respecto al surtidor, para lo cual es necesario que conduzca T_L. El borrado consiste en eliminar los portadores aplicando la tensión antes indicada en sentido contrario. Una vez programados los transistores T_p, la lectura se realiza seleccionando una posición mediante los dos decodificadores. De esta forma conducen los T_L de una fila y el T_C de la columna correspondiente. Sólo si el T_p que conecta el T_C con el T_L está grabado (posee portadores de carga), a la salida aparece un cero y, en caso contrario, aparece un uno. La tensión de salida del decodificador durante la lectura no es suficente para alterar el estado de la puerta aislada.

d) Utilizando transistores como los del apartado *c*) que poseen una puerta flotante en la que se pueden inyectar o extraer electrones mediante impulsos eléctri-

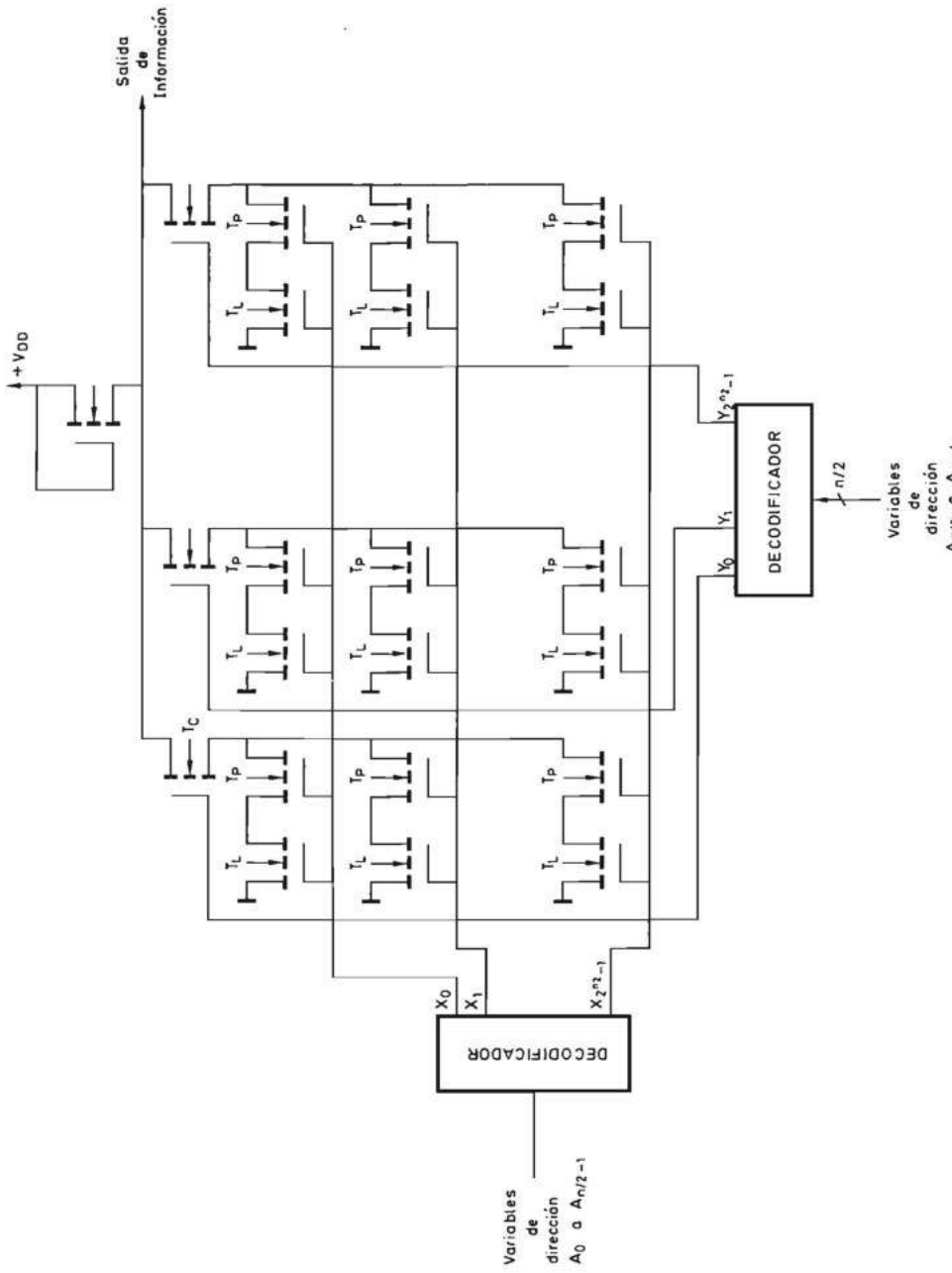


FIGURA 7.111.—**Esquema básico de una memoria pasiva reprogramable EEPROM.**

cos. La unidad de memoria se realiza exclusivamente con estos transistores, sin transistores de selección en serie y, por lo tanto, posee la característica de que el borrado se realiza en todos los transistores simultáneamente, por lo cual reciben el nombre de memorias flash (Flash memories).

Estas memorias poseen la misma densidad que las EPROM, con la ventaja de que pueden ser programadas en el propio circuito en el que se utilizan. En el apartado 7.2.3.1.3.2 se estudian las memorias Flash desde un punto de vista funcional y se indican sus aplicaciones.

7.3.3 Tecnología de las memorias de acceso serie

Estudiaremos fundamentalmente los registros de desplazamiento porque los otros tipos pueden ser realizados con ellos tal como se estudió en el apartado 7.2.3.2. En dicho apartado se vio que los registros de desplazamiento pueden ser estáticos o dinámicos. Los registros de desplazamiento estáticos están constituidos, en general, por biestables activados por flancos, interconectados entre sí de acuerdo con la estructura descrita en el apartado 6.3.5.2.3.

Los registros de desplazamiento dinámicos se han realizado en diversas tecnologías, pero la que ha adquirido una mayor importancia por la reducida complejidad de sus células básicas es la de MOS dinámico.

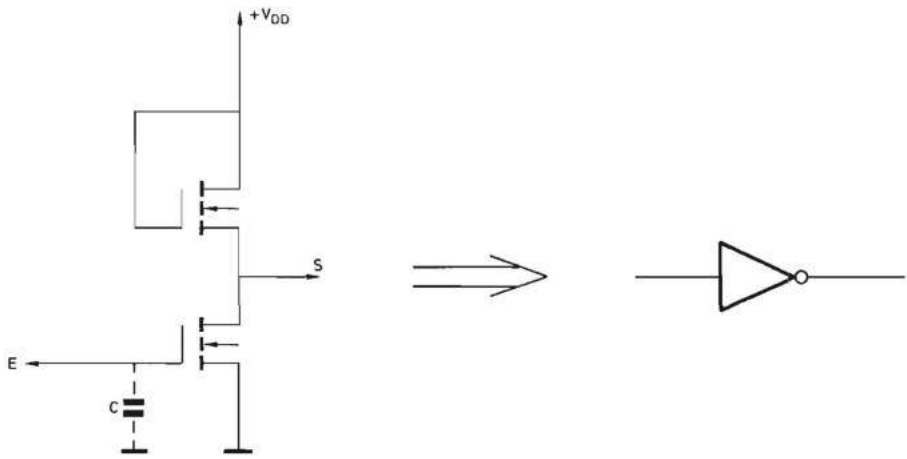


FIGURA 7.112.—**Inversor en tecnología N MOS.**

Las células de los registros de desplazamiento dinámicos en tecnología MOS están basadas en el mismo principio físico que las de las memorias de acceso aleatorio dinámicas descritas en el apartado 7.3.2.1, es decir, la capacidad de la puerta con respecto al surtidor de un transistor MOS.

Existen diferentes formas de realizar la célula básica de un registro de desplazamiento. Estudiaremos una de las más sencillas, la representada en la figura 7.113. Para un estudio más detallado se remite al lector a la bibliografía [INTE 75], [NATI 76].

El elemento fundamental de la célula MOS de un registro de desplazamiento es el inversor estático representado en la figura 5.56 o el inversor dinámico mostrado en la figura 5.60. Representaremos el inversor MOS independientemente de como esté realizado por el símbolo normalizado de inversión, tal como se indica en la figura 7.112. En esta figura se observa que entre la puerta y el surtidor del transistor MOS existe una capacidad parásita que puede almacenar temporalmente la información.

En la figura 7.113*a* se representa el esquema de la célula básica de un registro de desplazamiento dinámico. Se supone que los transistores son de canal *N* enriquecidos (enhancement).

La célula está formada por dos inversores acoplados entre sí por medio de un transistor MOS, al igual que el primer inversor y la entrada de información. El

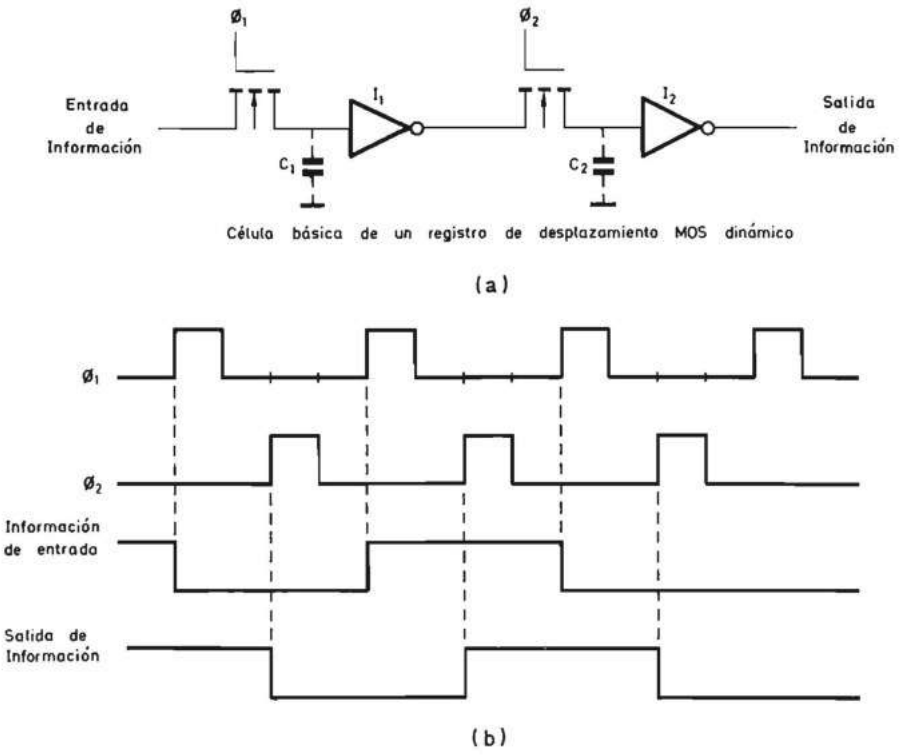


FIGURA 7.113.—**Célula básica de un registro de desplazamiento dinámico: *a*) Esquema del circuito; *b*) Diagrama de evolución temporal de sus señales.**

inversor MOS se puede considerar como una memoria de un bit que se graba o borra por nivel y almacena la información durante un cierto tiempo. Para que la información presente a la entrada aparezca a la salida con un retraso controlable mediante un generador de impulsos, las señales de control $\varnothing_1$ y $\varnothing_2$ de los transistores T_1 y T_2 no pueden estar activadas (llevar al transistor al estado de conducción) simultáneamente. En la figura 7.113*b* se representa el diagrama de secuencia de las señales $\varnothing_1$ y $\varnothing_2$ y la entrada y salida de información. Se supone que la información en la entrada cambia en fase con la activación de $\varnothing_1$, lo cual se produce normalmente porque el generador de impulsos que hace que se desplace la información en el interior de la memoria es el mismo que sincroniza el sistema digital al cual va acoplada la memoria.

Al activarse $\varnothing_1$, la información se almacena en C_1 y permanece almacenada en él cuando se desactiva $\varnothing_1$. Al activarse $\varnothing_2$, el estado de la salida del inversor I_1 (que es el inverso del memorizado en C_1) se acopla a C_2 y aparece invertido a la salida de la célula. Por lo tanto, la información de la salida de la célula es igual a la de entrada retrasada un tiempo t igual a la separación entre los flancos de subida de $\varnothing_1$ y $\varnothing_2$, tal como se observa en la figura 7.113*b*. Acoplando en serie

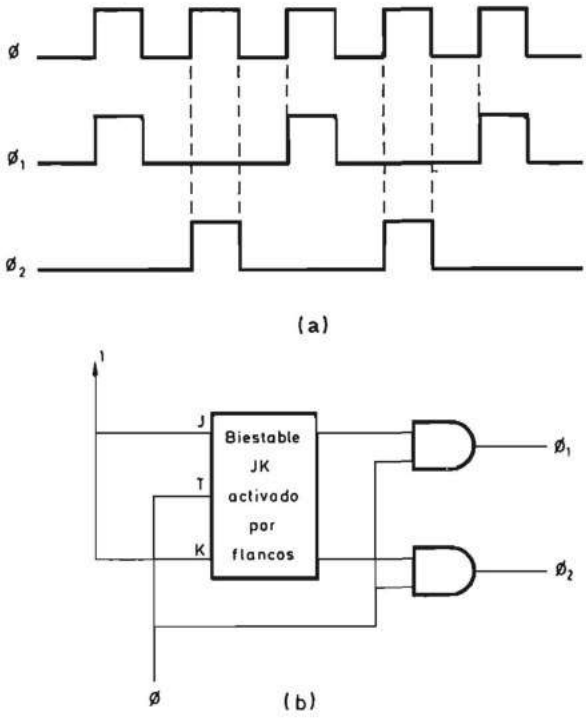


FIGURA 7.114.—**Señales de control de un registro de desplazamiento dinámico: *a*) Diagrama de evolución temporal; *b*) Esquema del circuito generador.**

n células como la que acabamos de estudiar se obtiene un retardo *nt* y se constituye un registro de desplazamiento dinámico.

Las señales $\varnothing_1$ y $\varnothing_2$ se pueden obtener a partir de un generador de impulsos $\varnothing$ tal como se indica en la figura 7.114*a*. El circuito necesario resulta muy sencillo y se representa en la figura 7.114*b*. El lector debe observar que el biestable *J-K* está montado como divisor por dos y deducir por sí mismo las señales $\varnothing_1$ y $\varnothing_2$ a partir de $\varnothing$. Se observa que la separación entre los flancos de subida de $\varnothing_1$ y $\varnothing_2$ es igual a un ciclo de $\varnothing$.

Los registros de desplazamiento han sido realizados como bloques funcionales en tecnología MOS de canal *P*, canal *N* y CMOS. En muchos de ellos el circuito que genera $\varnothing_1$ y $\varnothing_2$ a partir de $\varnothing$ está incluido en el interior del propio bloque funcional.

BIBLIOGRAFIA

[INTE 76] Memory design handbook. Intel Corporation. 1976.
[INTE 81] Intel Component Data Catalog. 2816 Electrically Erasable PROM data sheets. pp 2.23 a 2.34. Intel Corporation. 1981.
[INTE 86*a*] Memory components data book. Intel Corporation. 1986.
[INTE 86*b*] Integrated Device Technology Inc. High performance CMOS databook. 1986.
[INTE 90] Memory components handbook. Intel Corporation. 1990.
[NATI 76] Memory data book. National Semiconductor Corporation. 1976.
[NEC 87] Memory products data book. NEC. 1987.
[RIO 90] A. del Río. Controladores de TRC. Descripción y clasificación. Mundo Electrónico. Nº 207. Junio 1990.
[SEEQ 87] Seeq Technology Incorporated. Data book, 1987.
[TEXA 89] Mos memory data book. Texas Instruments. 1989.

Apéndice 1

Símbolos lógicos normalizados

A1.1 INTRODUCCION

Es un hecho constatado el vertiginoso desarrollo de la electrónica digital en la década de los años setenta, que produjo la aparición de bloques funcionales digitales en circuito integrado cada vez más complejos, que hicieron necesaria la creación de símbolos adecuados para su representación. Dicha creación se realizó de forma paulatina conforme el progreso de la capacidad de integración iba generando nuevos bloques funcionales, pero el proceso careció de un cuerpo de doctrina que facilitase la representación de nuevos bloques funcionales.

Por ello diversos organismos normalizadores iniciaron a finales de los años setenta los trabajos de desarrollo de un conjunto de normas para la representación gráfica de las funciones lógicas, que dio como resultado la creación de símbolos lógicos normalizados adoptados por la Comisión Electrotécnica Internacional (International Electrotechnical Commission) que forma parte de la ISO (International Standard Organization).

La adopción de estos símbolos de forma generalizada, y muy especialmente por los fabricantes de circuitos integrados en sus manuales, hará más fácil la interpretación de los esquemas de los sistemas digitales y contribuirá a potenciar las técnicas de diseño asistido por computador (CAD) de sistemas electrónicos digitales.

Por todo ello, el ingeniero especializado en electrónica necesita realizar el esfuerzo de asimilar la nueva simbología y utilizarla en los esquemas de los circuitos desarrollados por él.

En sucesivos apartados se analizan los nuevos símbolos normalizados, comparándolos con los más utilizados hasta ahora y a los cuales sustituyen.

A1.2 SIMBOLO GRAFICO NORMALIZADO

La norma de representación de los bloques funcionales digitales adoptada por la Comisión Electrotécnica Internacional está basada fundamentalmente en la utilización de un rectángulo orientado verticalmente al que se añaden los símbolos adecuados para expresar gráficamente, sin necesidad de ningún texto explicativo, la función que realiza.

En la figura A1.1 se representa el rectángulo general normalizado, en cuya parte superior se coloca el símbolo de descripción general, que indica el tipo de función que realiza.

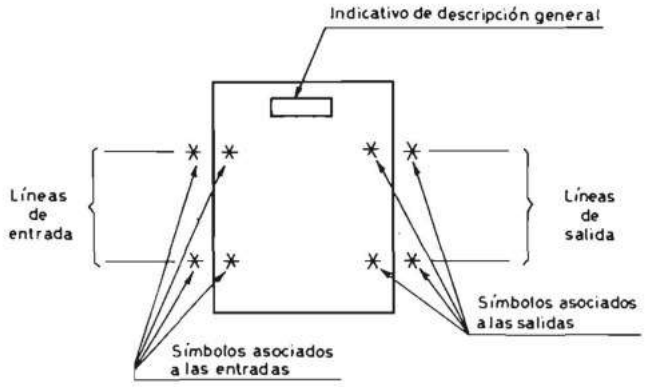


FIGURA A1.1.—**Composición del símbolo lógico gráfico normalizado.**

A la izquierda del símbolo gráfico se suelen colocar las líneas de entrada y a la derecha las de salida, de tal manera que el flujo de información se produce desde la izquierda hacia la derecha.

Cada entrada puede llevar uno o dos símbolos asociados que especifican la forma en que actúa sobre el circuito.

De igual forma, cada salida puede llevar también símbolos asociados que especifican sus características. En sucesivos apartados se estudian los diferentes símbolos asociados a las entradas y a las salidas.

Cuando un bloque funcional complejo está compuesto a su vez por varios bloques funcionales idénticos, se puede representar mediante símbolos gráficos contiguos, colocando el símbolo de descripción general solamente en el superior. En la figura A1.2 se representan cuatro funciones digitales idénticas de dos variables de entrada y una de salida contenidas en un único circuito integrado.

A veces, los bloques funcionales complejos están compuestos por varios circui-

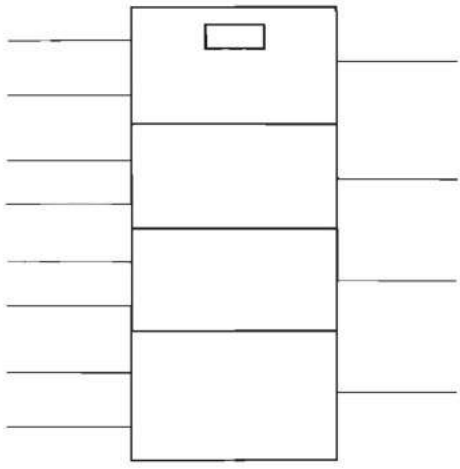


FIGURA A1.2.—**Bloque funcional con cuatro funciones digitales idénticas.**

tos idénticos que poseen entradas específicas para cada uno de ellos y entradas comunes a todos.

En estos casos, muy usuales (un ejemplo característico es el bloque multiplexor múltiple), se le añade a los símbolos rectangulares contiguos un bloque común como el representado en la parte superior de la figura A1.3*a*, al que se conectan

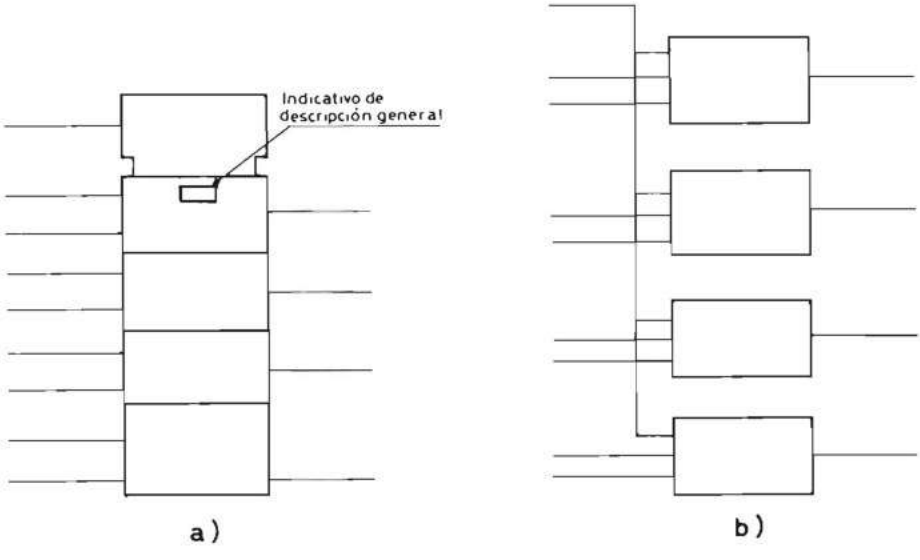


FIGURA A1.3.—*a*) **Símbolo de un bloque funcional con cuatro funciones digitales que poseen una entrada común.** *b*) **Circuito equivalente del símbolo representado en** *a*).

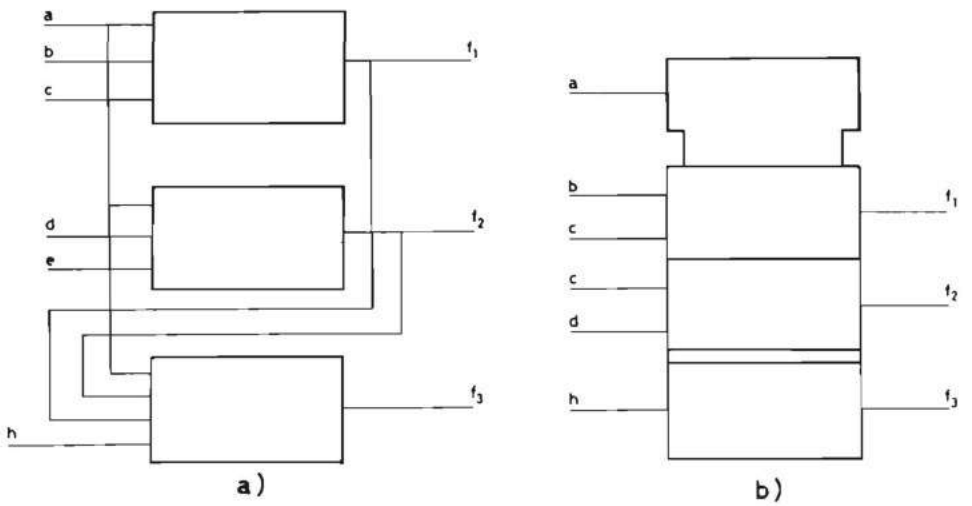


FIGURA A1.4.—*a*) **Bloque funcional con dos funciones independientes y una que depende de las primeras.** *b*) **Símbolo gráfico normalizado del bloque representado en a).**

todas las entradas comunes al resto. En la figura A1.3*a* se representa el símbolo gráfico de un bloque funcional con cuatro funciones digitales idénticas que poseen dos entradas independientes cada una y una común a todas. En la figura A1.3*b* se representa el circuito al que equivale el símbolo de la figura A1.3*a*.

Algunos bloques funcionales están compuestos por varias funciones digitales independientes y otra que depende de las salidas de las primeras, tal como se representa en el ejemplo de la figura A1.4*a*. Es posible simplificar la representación tal como se indica en la figura A1.4*b*.

Dado que la variable *a* es común a las tres funciones, se une al bloque común superior. Para indicar que la salida f_3 del bloque inferior depende de la variable

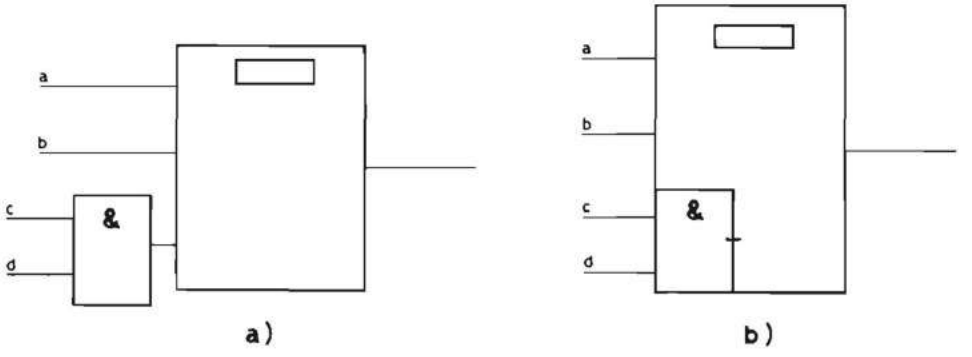


FIGURA A1.5.—*a*) **Funciones lógicas interconectadas.** *b*) **Símbolo normalizado equivalente.**

h y de las salidas f_1 y f_2 de los otros dos, se le separa del anterior mediante dos segmentos paralelos y se indica solamente la variable h.

En ocasiones, parte de las entradas de un bloque funcional se combinan por medio de una determinada función lógica diferente de la que realiza el citado bloque. Una posible representación es la indicada en el ejemplo de la figura A1.5*a*, en la que el producto lógico de las variables c y d se aplica al segundo bloque. En la figura A1.5*b*, se indica una representación simplificada en la que se encaja un símbolo en el otro (embedded symbols).

A1.3 SIMBOLOS ASOCIADOS A LAS ENTRADAS Y A LAS SALIDAS

En el apartado anterior se analizó el símbolo general utilizado para representar los distintos elementos lógicos. El proceso de creación de un conjunto de símbolos normalizados necesita, además, de la adecuada descripción de los distintos tipos de terminales y de la forma en que las señales presentes en ellos actúan sobre el circuito.

FIGURA A1.6.—**Símbolo normalizado de inversión en entradas *a*) y salidas *b*).**

Los símbolos asociados a los terminales de los elementos lógicos pueden hacer referencia a la función lógica que realizan o a sus características eléctricas.

La función lógica de un terminal puede ser específica de un determinado tipo de bloque funcional o idéntica para todos los bloques.

Las funciones específicas se estudian en apartados sucesivos y se analizan seguidamente las de uso general.

En la figura A1.6 se representan los símbolos de inversión asociados a las entradas y a las salidas que consisten en un círculo. En el caso de una entrada (fig. A1.6*a*), el círculo indica que un cero externo se convierte en un uno aplicado al bloque y viceversa.

De forma similar, un círculo a la salida de un bloque funcional (fig. A1.6*b*) indica que la función realizada por el bloque se invierte antes de ser conectada a la salida.

El círculo utilizado en la figura A1.6 para representar la inversión no indica si la lógica utilizada es positiva o negativa. Por ello se ha normalizado un triángulo para representar la inversión en lógica positiva, tal como se ilustra en la figura A1.7.

FIGURA A1.7.—**Símbolo normalizado de inversión en lógica positiva: *a*) entrada *b*) salida.**

En los sistemas secuenciales existen entradas que actúan por nivel (como, por ejemplo, la de inhibición, las de selección de modo de operación, etc.) y otras que actúan por flanco o cambio de nivel (como, por ejemplo, una entrada de contaje). Las primeras utilizan los símbolos de la figura A1.6 para representar la inversión o, bien, carecen de símbolo en el caso de que no exista la inversión.

Para distinguir las entradas que actúan por flanco o dinámicas de las que lo hacen por nivel o estáticas, se añade a las primeras un símbolo interno que consiste en un triángulo isósceles. En la figura A1.8*a* se representa una entrada que actúa mediante cambios de cero a uno y en la A1.8*b* otra que actúa mediante cambios de uno a cero.

En el supuesto de lógica positiva, el círculo se puede sustituir por un triángulo tal como se indica en la figura A1.9.

Algunas entradas de bloques funcionales poseen histéresis, es decir, que los cambios de cero a uno y uno a cero se realizan a distinto nivel. Estas entradas se indican mediante el símbolo interno representado en la figura A1.10.

FIGURA A1.8.—**Entradas dinámicas:**
***a*) Entrada que actúa mediante los cambios o flancos de cero a uno.**
***b*) Entrada que actúa mediante los cambios o flancos de uno a cero.**

FIGURA A1.9.—**Entradas dinámicas en lógica positiva.**

En el caso de la salida, es además interesante indicar las características eléctricas.

Para ello se utilizan los símbolos internos indicados en las figuras A1.11 y A1.12. La primera de ellas representa el símbolo de la salida de tres estados cuyo circuito eléctrico con transistores bipolares se indica a la derecha.

La figura A1.12 representa la salida de circuito abierto que presenta a su vez dos variantes. La figura A1.13 representa una salida de colector abierto (L type open circuit) (drenador abierto en el caso de transistores MOS) y la figura A1.14 una salida de emisor abierto (H type open circuit) (surtidor abierto en el caso de transistores MOS). A su vez, las salidas de circuito abierto pueden llevar incorporada una resistencia en su interior que constituye una carga pasiva del transistor.

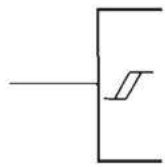


FIGURA A1.10.—**Entrada con histéresis.**

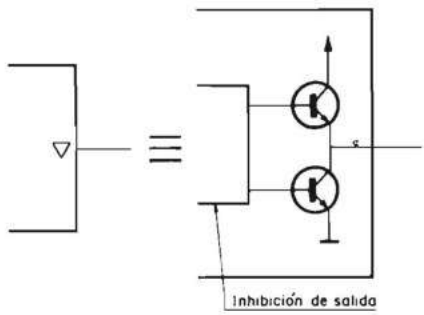


FIGURA A1.11.—**Símbolo de una salida de tres estados.**

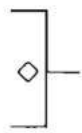


FIGURA A1.12.—**Símbolo de una salida de circuito abierto.**

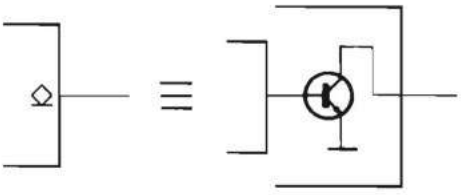


FIGURA A1.13.—**Símbolo de una salida de colector o drenador abierto.**

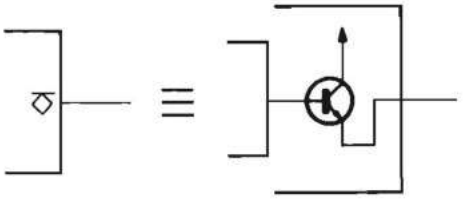


FIGURA A1.14.—**Símbolo de una salida de emisor o surtidor abierto.**

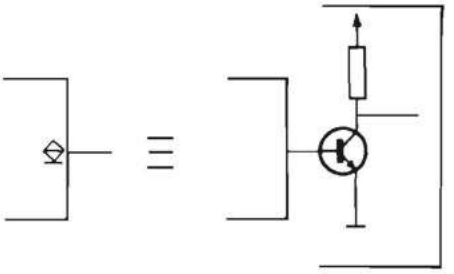


FIGURA A1.15.—**Símbolo de una salida con carga pasiva en el colector.**

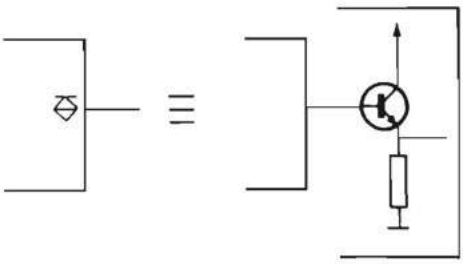


FIGURA A1.16.—**Símbolo de una salida con carga pasiva en el emisor.**

La figura A1.15 nos muestra una salida pasiva por colector (passive pull-up output), y la A1.16 ilustra una salida pasiva por emisor (passive pull-down output).

A1.4 REPRESENTACION NORMALIZADA DE LOS SISTEMAS COMBINACIONALES

A1.4.1 Generalidades

Los sistemas combinacionales se pueden clasificar en dos grandes categorías:

a) Aquellos en los que no existe ninguna relación de dependencia entre las entradas, es decir que el estado de ninguna entrada influye sobre la forma en que el circuito actúa sobre las demás entradas para obtener la salida. Estos sistemas realizan determinadas funciones lógicas y suelen recibir el nombre de *puertas*.

b) Aquellos en los que existe una relación de dependencia entre unas entradas y el resto. Ejemplo típico son las entradas de selección, las entradas de inhibición, etc. La realización física de estos sistemas da lugar a bloques funcionales complejos realizados en circuitos de escala de integración media (MSI) o que forman parte de sistemas más complejos realizados en circuitos de gran o muy gran escala de integración (LSI y VLSI).

En apartados sucesivos se analizan ambos tipos de sistemas combinacionales.

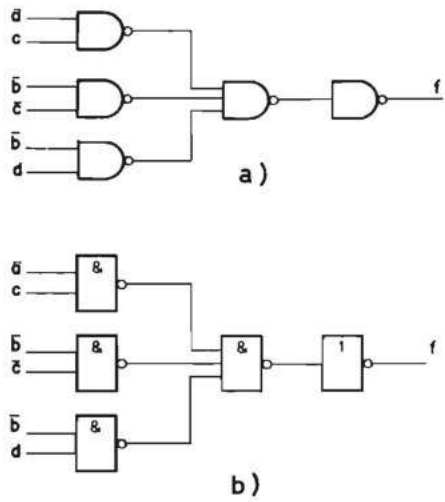


FIGURA A1.17.—**Ejemplo de circuito lógico realizado con puertas NO-Y(NAND): *a*) Símbolos no normalizados; *b*) Símbolos normalizados.**

A1.4.2 Puertas lógicas

El símbolo lógico utilizado es el rectángulo general, en cuya parte superior se coloca el indicativo de la función realizada.

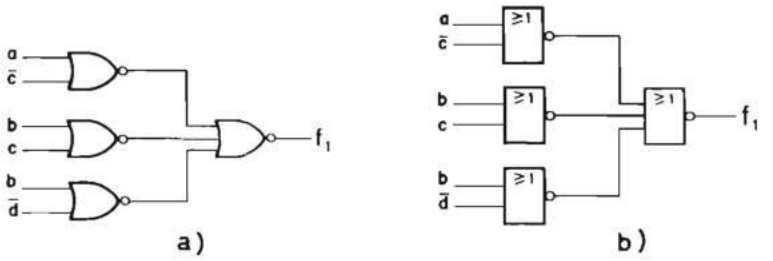


FIGURA A1.18.—**Ejemplo de circuito lógico realizado con puertas NO-O (NOR): *a*) Símbolos no normalizados; *b*) Símbolos normalizados.**

En la tabla A1.1 se representan los indicativos o símbolos de las funciones lógicas más utilizadas. En las figuras A1.17 y A1.18 se ilustran los esquemas de sendos circuitos combinacionales realizados con puertas NO-Y (NAND) y NO-O (NOR) respectivamente, con los símbolos normalizados *b*) y con los utilizados hasta ahora *a*).

Símbolo	función lógica
&	Puerta o Función Y
≥ 1	Puerta o función O
= 1	Puerta O-exclusiva de dos entradas: su salida vale uno si sólo una de las entradas se encuentra en ese nivel.
=	Función lógica igualdad = Todas las entradas deben estar en el mismo estado para que a la salida haya un uno lógico.
$=m$	m y sólo m entradas deben estar a nivel uno para que a la salida haya un uno.
$>n/2$	La salida está a nivel uno si la mayoría (más de la mitad) de las entradas están en ese nivel.
$\geq m$	Función umbral: su salida se pone a nivel uno si un mínimo de m entradas se encuentran en ese nivel.
$2k$	La salida está en el nivel uno siempre y cuando haya un número par de entradas en este estado.
$2k+1$	Idem con un número impar de entradas.
▷ o ◁	Elemento amplificador (buffer), es decir, elemento con capacidad de salida mayor de lo usual. El símbolo señala el sentido del flujo de la señal.

TABLA A1.1.—**Indicativos generales de las funciones lógicas.**

A1.4.3 Símbolos de bloques funcionales combinacionales complejos

Los bloques funcionales combinacionales son circuitos combinacionales complejos en los que existen, en general, determinadas entradas que modifican la forma en que el circuito actúa sobre las demás para obtener las variables de salida. Las primeras pueden recibir el nombre genérico de entradas de control y las segundas entradas de información u operativas. Se dice que existe entre ambos tipos de entradas una relación de dependencia. Hay varios tipos de dependencias que se distinguen por la forma en que se realiza la citada modificación, pero existe una regla general, común a todas, que es la siguiente:

«La entrada que modifica la forma en que el circuito actúa sobre las de información u operativas se indica con una letra correspondiente al tipo de dependencia seguida de un número. Cada una de las entradas afectadas tiene asignado el mismo número».

De igual forma, en los circuitos combinacionales complejos existen entradas que modifican el resultado o variables de salida del propio circuito a través de una relación de dependencia, a la que se aplica la misma regla anterior.

Esta regla se comprende fácilmente mediante varios ejemplos que se analizan, al mismo tiempo que los diferentes tipos de dependencias, en sucesivos apartados.

Cada tipo de bloque funcional tiene asignado un indicativo de descripción general de acuerdo con la función que realiza. En la tabla A1.2 se presentan los indi-

Símbolo	*Bloque funcional*
X/Y	Convertidor de código (por ejemplo, de BCD a siete segmentos).
HIPRI/BIN	Codificador con prioridad en binario natural.
COMP	Comparador.
MUX	Multiplexor.
DMUX o DX	Demultiplexor.
Σ	Sumador (Adder)
P—Q	Restador (Subtractor)
CPG	Generador de acarreo anticipado (Carry propagation generator)
ALU	Unidad aritmética y lógica (Arithmetic logic unit)
Π	Multiplicador (Multiplier)
ROM	Memoria totalmente pasiva (Read only memory)
PROM	Memoria pasiva programable (Programmable read only memory).
EPROM	Memoria pasiva reprogramable (Erasable programmable read only memory).
PLA	Matriz lógica programable (Programmable logic array)
PAL	Matriz lógica Y-programable (Programmable array logic or Programmable and array logic)

TABLA A1.2.—**Indicativos generales de los bloques funcionales combinacionales.**

cativos correspondientes a los principales bloques funcionales combinacionales entre los que se incluyen las memorias de acceso aleatorio pasivas (ROM, PROM y RPROM), las matrices lógicas programables [Programmable logic array (PLA)] y las matrices lógicas Y-programables [Programmable array logic (PAL)].

Seguidamente se analizan los principales tipos de dependencias y la forma de combinarlas.

A1.4.3.1 Relación de dependencia Y [G(AND)]

Como su nombre expresa, representa un producto lógico entre la entrada indicada mediante la letra *G* seguida de un símbolo numérico α y las entradas o salidas de información indicadas con el símbolo numérico α.

En la figura A1.19*a* se representa la función de la relación Y entre una entrada *n* y un bloque funcional, mediante la inclusión de una puerta Y. En la figura A1.19*b*, se indica el símbolo normalizado, asignando a la entrada *n* el indicativo $G\alpha$ y a la salida el indicativo α. Es conveniente resaltar que la figura A1.19 constituye sólo una forma gráfica de representar la relación de dependencia Y; en la práctica la función Y no necesita una puerta independiente sino que se realiza conjuntamente con las demás funciones del bloque.

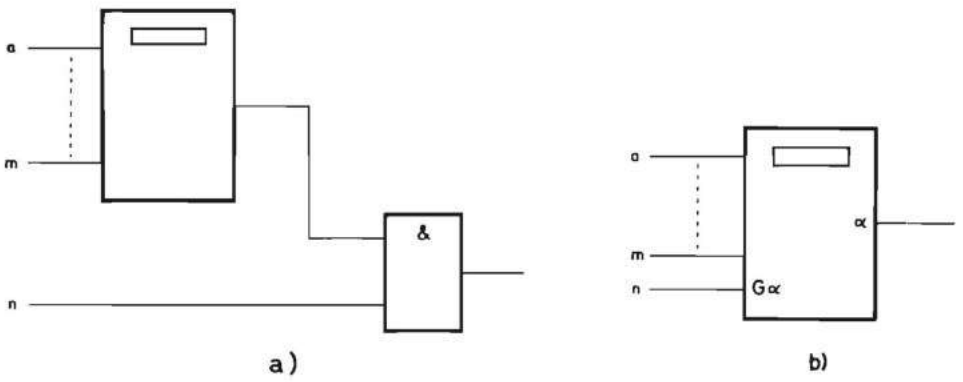


FIGURA A1.19.—**Relación de dependencia Y entre una entrada y una salida.**

Un ejemplo típico de la dependencia Y entre entradas y salidas lo constituyen los demultiplexores. En la figura A1.20 se representa un demultiplexor de 8 canales de salida. A las entradas 0, 1 y 2, se les asignan los símbolos $G\frac{0}{7}$ que indican que cada combinación decimal 0 a 7 de las citadas variables actúa sobre la salida 0 a 7 respectiva de acuerdo con la función Y (*G*).

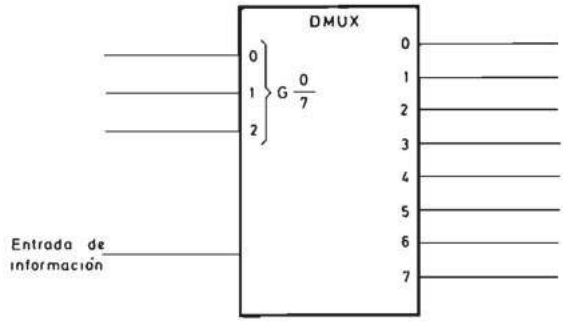


FIGURA A1.20.—**Símbolo de un demultiplexor de 8 canales de salida.**

En la figura A1.21 se representa la función de la relación Y entre una entrada y otra entrada, que por su similitud con la misma relación entre una entrada y una salida no necesita explicación. Un ejemplo típico de esta relación lo constituyen los multiplexores. En la figura A1.22 se representa un multiplexor de cuatro canales en el que cada combinación de las variables de selección 0 y 1 selecciona uno

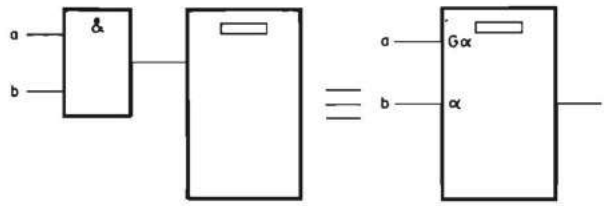


FIGURA A1.21.—**Relación de dependencia Y entre una entrada y otra entrada.**

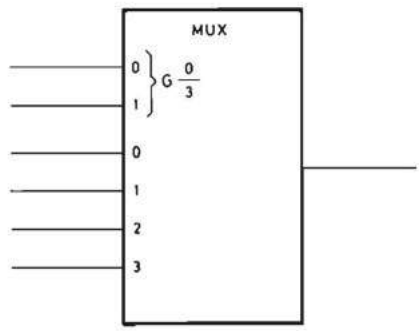


FIGURA A1.22.—**Símbolo de un multiplexor de cuatro canales.**

de los cuatro canales de entrada. Para indicarlo, a estas variables se les asigna el símbolo $G\frac{0}{3}$ y a cada canal de entrada el número correspondiente 0, 1, 2 y 3.

Se puede resumir, por lo tanto, y definir la relación de dependencia *Y* diciendo:

Cuando una entrada $G\alpha$ se encuentra en su estado interno (el que existe en el interior del bloque sin tener en cuenta una posible inversión externa) cero, todas las entradas y salidas internas indicadas mediante el símbolo α se ponen a cero. Si, por el contrario, $G\alpha$ se encuentra en nivel uno, todas las entradas internas afectadas por ella se encuentran en el nivel lógico fijado por los terminales de entrada y las salidas al nivel fijado por las entradas y la función lógica del bloque.

A1.4.3.2 Relación de dependencia O [V(OR)]

Esta relación es similar a la anterior sustituyendo la función Y por la O.

Por ello cuando una entrada $V\alpha$ se encuentra en nivel uno, todas las entradas y salidas internas indicadas mediante el símbolo α se ponen a uno. Si, por el contrario, $V\alpha$ se encuentra en nivel cero, todas las entradas afectadas por ella se encuentran en el nivel lógico fijado por los terminales de entrada y las salidas al nivel fijado por las entradas y la función lógica del bloque.

A1.4.3.3 Relación de inversión [N(Negate)]

Esta relación representa una función O-exclusiva entre las entradas indicadas con la letra *N* seguida de un símbolo numérico α y las entradas o salidas de información indicadas con el mismo símbolo numérico α.

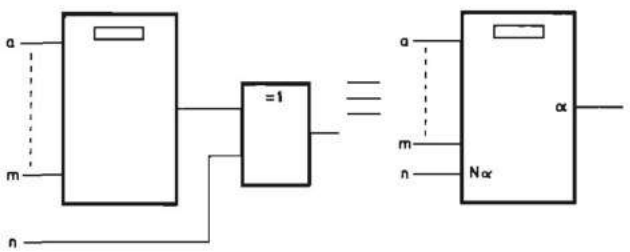


FIGURA A1.23.—**Relación de dependencia inversión (*N*) entre una entrada y una salida.**

En la figura A1.23 se representa esta función entre una entrada y una salida y su indicación con la relación de dependencia *N*.

La figura A1.24 representa un ejemplo de la relación de dependencia *N*, mediante el multiplexor de cuatro canales de la figura A1.22, al que se le ha añadido

una entrada que actúa sobre la salida con la relación *N*. Si la entrada *N*4 se encuentra en nivel cero, la salida 4 adopta el nivel que le corresponde según el estado de las restantes entradas. Si *N*4 se encuentra en nivel uno, la salida 4 adopta el

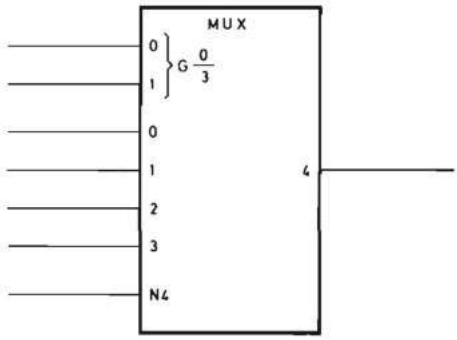


FIGURA A1.24.—**Símbolo de un multiplexor de cuatro canales con una entrada de control de inversión de la salida.**

nivel opuesto al anterior. Por lo tanto, *N*4 constituye una variable de control de la inversión de la salida del multiplexor.

El lector puede obtener sin dificultad la definición de la relación *N*.

A1.4.3.4 Relación de desinhibición [EN(Enable)/Inhibición]

Como su nombre expresa, cuando una entrada tiene la denominación *EN*α actúa sobre las entradas que poseen el símbolo α inhibiendo su acción, si se encuentra en nivel cero y desinhibiéndola si se encuentra en nivel uno. Si *EN*α actúa sobre las salidas α, dependiendo de la tecnología, puede realizar una de las tres funciones:

a) Si las salidas son de colector abierto, corta el transistor correspondiente.
b) Si las salidas son de tres estados, las coloca en estado de alta impedancia.
c) Si son salidas normales, se ponen en el estado interno cero, lo cual es idéntico a la relación *G*.

Cuando una entrada *EN* no lleva ningún número α, actúa sobre todas las salidas del bloque de la forma indicada anteriormente. Esta función se utiliza mucho, tanto en bloques funcionales combinacionales como secuenciales.

En la figura A1.25 se representa un multiplexor de cuatro canales de entrada con salida de tres estados y entrada de desinhibición *EN*.

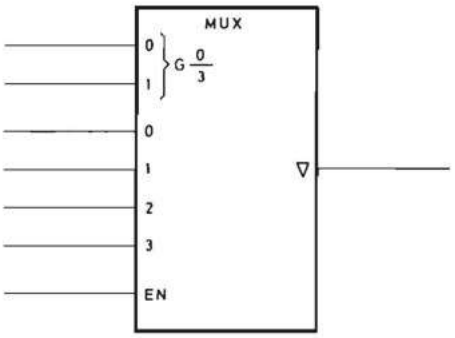


FIGURA A1.25.—**Símbolo de un multiplexor de cuatro canales con una entrada de desinhibición (*EN*) de la salida de tres estados.**

A1.4.3.5 Relación de modo de operación M (Mode)

Las entradas de tipo *M* pueden afectar a entradas o salidas de información.

Este tipo de relación es especialmente interesante en los bloques funcionales secuenciales que se estudian en el apartado A1.5.3, pero también se aplica a los combinacionales.

La relación *M* afecta a las entradas especialmente en los sistemas secuenciales y por ello se remite al lector al apartado A1.5.3 de este apéndice.

La relación *M* afecta a las salidas de los sistemas combinacionales seleccionando su modo de operación. En la figura A1.26 se representa un ejemplo constituido por cuatro amplificadores (drivers) con una entrada de desinhibición [enable (*EN*)]

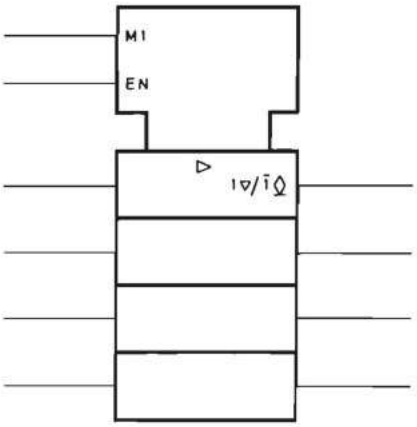


FIGURA A1.26.—**Cuádruple amplificador (driver) con entradas de modo de operación (*M1*) y de desinhibición (*EN*).**

y una entrada de modo de operación *M*1 comunes. La entrada *M*1 selecciona el modo de operación de la salida de tal manera que si se encuentra en nivel uno la salida es de tres estados y si se encuentra en nivel cero es de colector abierto. Esto se indica asignándole a la salida el conjunto de símbolos $1\nabla/\bar{1}\,\Diamond$.

A1.4.3.6 Relación de conexión (Z)

Esta relación indica que existe una conexión entre el terminal definido por $Z\alpha$ y el terminal α y su finalidad es reducir la complejidad de la representación de bloques funcionales complejos.

En la figura A1.27 se representa un ejemplo. Se trata de un sistema combinacional que, además de realizar una determinada función con sus variables de entrada, realiza la función Y de cuatro de ellas. Para no tener que representar expresamente las conexiones, a las entradas se las denomina *Z*1 a *Z*4 y a las entradas de la puerta Y se las indica mediante los números 1 a 4.

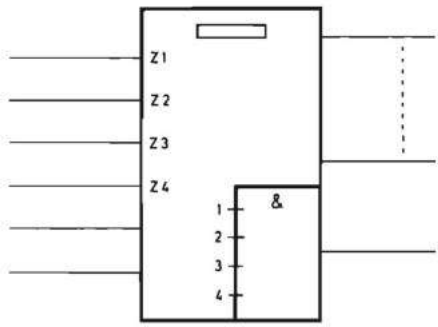


FIGURA A1.27.—**Ejemplo de relación de conexión (*Z*).**

A1.4.3.7 Relación de direccionamiento (Address) A

Esta relación se utiliza para representar la selección de un elemento lógico entre varios que constituyen un conjunto multidimensional. El caso más utilizado son las memorias de acceso aleatorio que poseen un cierto número de posiciones de un determinado número de bits cada una.

La selección se puede realizar de dos formas:

a) Mediante un terminal independiente para cada posición.
b) Mediante un cierto número n de terminales tal que 2^n es igual al número total de posiciones.

La primera se utiliza cuando la memoria posee un número pequeño de posiciones (en general, no superior a 8). La segunda es la más utilizada en la práctica.

En la figura A1.28 se representa una memoria de acceso aleatorio pasiva programable (PROM) de cuatro posiciones de 8 bits con salida de tres estados (controlada mediante la entrada de desinhibición *EN*), con cuatro terminales de dirección *A*1, *A*2, *A*3 y *A*4.

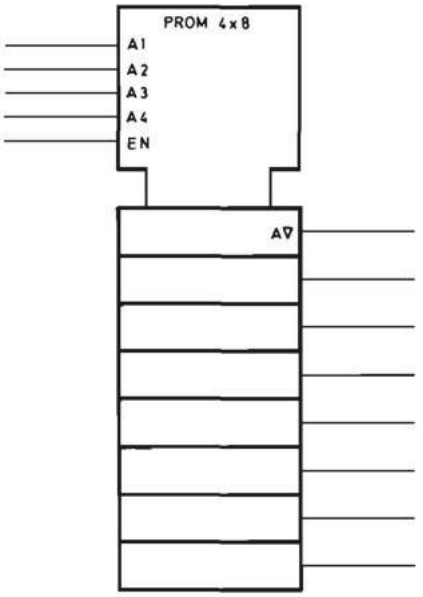


FIGURA A1.28.—**Símbolo de una memoria de acceso aleatorio pasiva programable (PROM) de cuatro posiciones de 8 bits con un terminal de dirección por posición.**

A la salida de la memoria aparece, si está desinhibida ($EN = 1$), la información de la posición correspondiente al terminal *Ai* que se encuentra en nivel uno (simultáneamente sólo una entrada *A* puede estar en dicho estado). La letra *A* colocada en la salida indica que la información que aparece en ella se selecciona mediante las entradas *Ai*.

En la figura A1.29 se representa una memoria del mismo número de posiciones y bits por posición que la representada en la figura A1.28, que se diferencia de ella en que la selección se realiza mediante dos bits ($2^2 = 4$), de tal manera que en cada instante se selecciona la posición correspondiente a la combinación presente en dichos bits.

A1.4.3.8 Combinación de relaciones de dependencia

Las relaciones de dependencia se pueden combinar entre sí, afectando conjuntamente a la forma de operación de un bloque funcional.

En la práctica, dicha combinación se puede realizar de dos formas principales:

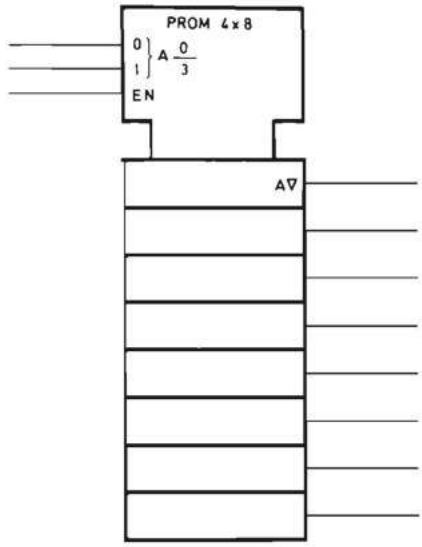


FIGURA A1.29.—**Símbolo de una memoria de acceso aleatorio pasiva programable (PROM) de cuatro posiciones de 8 bits con dos terminales ($2^2 = 4$) de dirección.**

a) Combinación de simultaneidad.

Se especifica mediante los símbolos adecuados separados por comas. Se presenta cuando una entrada que actúa de una sola forma sobre el contenido de un bloque funcional, está condicionada por el estado de varias entradas simultáneamente, o cuando el estado de una salida depende de varias entradas simultáneamente.

En la figura A1.30 se representa un bloque funcional combinacional que posee entrada de control de tercer estado *EN*2 y de desinhibición *G*1 que afectan simultá-

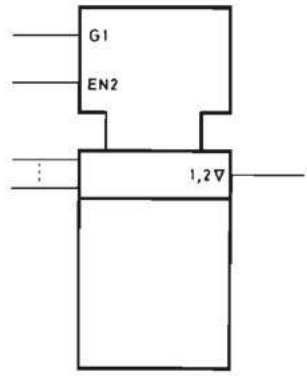


FIGURA A1.30.—**Combinación de simultaneidad de relaciones de dependencia.**

A1.5.3 Relaciones de dependencia en sistemas secuenciales

En los sistemas secuenciales se aplican también las relaciones de dependencia estudiadas en el apartado A1.4.3 para los sistemas combinacionales: la relación Y [*G*(AND)], la relación O [*V*(OR)], la relación inversión [*N*(NEGATE)], la relación de desinhibición/inhibición [*EN*(Enable)], la relación de modo de operación [*M*(Mode)], la relación de conexión (*Z*), y la de direccionamiento [*A*(Address)].

Indicativo de entrada o salida	*Función*
J y *K*	Entradas de un biestable *J-K*.
D	Entrada de un biestable *D* y en general entrada de información a cualquier bloque funcional secuencial.
C	Entrada de control de un sistema secuencial o entrada de impulsos de sincronismo.
S	Entrada de puesta a uno (Set) de un biestable o de un bloque funcional secuencial.
R	Entrada de puesta a cero (Reset) de un biestable o de un bloque funcional secuencial.
→	Entrada de impulsos de desplazamiento hacia la derecha.
←	Entrada de impulsos de desplazamiento hacia la izquierda.
+	Entrada de impulsos de contaje ascendente.
—	Entrada de impulsos de contaje descendente.
CT = *m*	Entrada de contenido que, cuando se pone en nivel uno hace que el contenido de los elementos de memoria de un bloque funcional secuencial sea igual al número binario equivalente al decimal *m*.
CT = *m*	Salida de contenido que, cuando se encuentra en nivel uno indica que el contenido de los elementos de memoria de un bloque funcional secuencial, es igual al número binario equivalente al decimal *m*.

TABLA A1.4.—**Indicativos de entradas y salidas de los sistemas secuenciales.**

Pero en los sistemas secuenciales hay relaciones de dependencia específicas como son las de puesta a cero (*R*) y puesta a uno (*S*) y la de control (*C*). Además la relación de modo de operación (*M*) presenta aspectos diferenciados con relación a los sistemas combinacionales.

Por ello en sucesivos apartados se estudian estos tipos de relaciones en sistemas secuenciales.

A1.5.3.1 Relaciones de puesta a cero (R) y puesta a uno (S)

Esta relación define la actuación de determinadas entradas de los bloques funcionales secuenciales que ponen en un cierto estado los elementos de memoria.

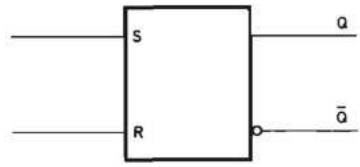


FIGURA A1.32.—**Diagrama de bloques del biestable *R-S*.**

La entrada de puesta a cero [*R*(Reset)] pone en estado cero los elementos de memoria y la entrada de puesta a uno [*S*(Set)] los pone en uno.

El ejemplo más significativo de esta relación de dependencia es el biestable asíncrono *R-S* [ver apartado 6.2.2.1], cuyo símbolo lógico se representa en la figura A1.32.

En la figura A1.33 se representa el símbolo lógico de un bloque funcional síncrono que posee una entrada de puesta a cero (*R*) de todos los elementos de memoria que lo constituyen y que se coloca en la parte común.

A1.5.3.2 Relación de control (C)

Esta relación de dependencia es específica de los elementos de memoria o biestables síncronos. Las entradas denominadas *C* controlan la actuación de las demás entradas, denominadas de información, sobre el estado del biestable. En los símbolos no normalizados esta entrada recibe la denominación de *T*(Trigger).

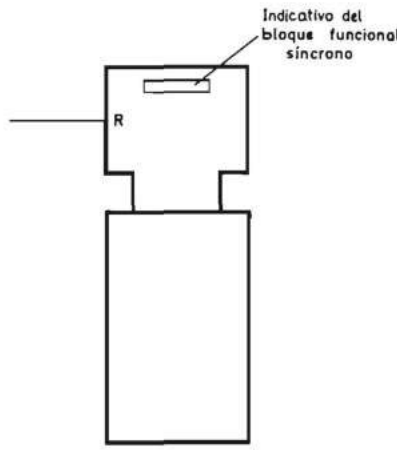


FIGURA A1.33.—**Símbolo lógico de un bloque funcional síncrono con entrada de puesta a cero (*R*) de todos los biestables.**

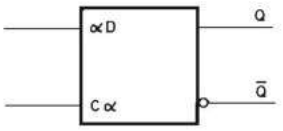


FIGURA A1.34.—**Símbolo del biestable *D* activado por niveles.**

Existen dos tipos de entradas *C* según la forma en que realizan el control:

a) De control por nivel, que cuando se encuentran en un determinado nivel hacen que las entradas de información actúen sobre el estado del biestable. Se denominan con la letra *C* seguida de un dígito decimal.

En la figura A1.34 se representa un biestable *D* activado por niveles. La entrada de sincronización recibe la denominación *C* seguida del dígito numérico α y la entrada de información *D* tiene asignada el mismo número α.

En la figura A1.35 se representa otro ejemplo de utilización de la relación de control *C*, constituido por un registro de *n* biestables *D* activados por niveles con una entrada de impulsos única.

b) De control por flanco, que hacen que las entradas de información actúen sobre el estado del biestable cuando se produce en ellas un cambio de nivel que se denomina activo y puede ser de cero a uno o de uno a cero. Estas entradas reciben también la denominación *C* seguida de un dígito decimal y precedida del indicativo de flanco, representado en la figura A1.8.

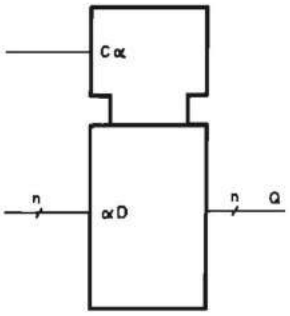


FIGURA A1.35.—**Símbolo de un registro de entrada y salida paralelo formado por *n* biestables *D* activados por niveles.**

En la figura A1.36 se representa un biestable *D* activado por flancos de subida y en la figura A1.37 un registro de *n* biestables *D* activados mediante una variable de control que actúa con los flancos de bajada.

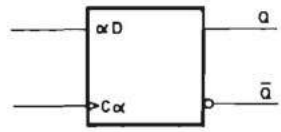


FIGURA A1.36.—**Símbolo del biestable *D* activado por flancos de subida.**

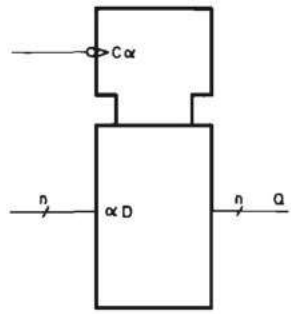


FIGURA A1.37.—**Símbolo de un registro de entrada y salida en paralelo formado por *n* biestables *D* activados por flancos de bajada.**

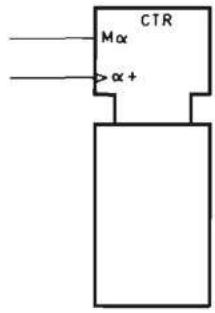


FIGURA A1.38.—**Símbolo de un contador síncrono de contaje ascendente con entrada de modo de operación Mα.**

En algunos bloques funcionales, como por ejemplo los contadores, no se pone el indicativo *C* porque está clara la función del terminal correspondiente mediante el indicativo de flanco (ver figuras A1.38 a A1.41).

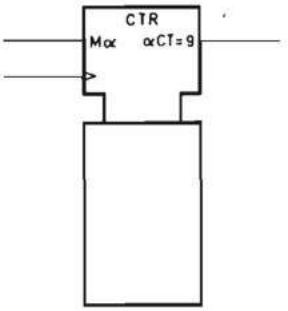


FIGURA A1.39.—**Símbolo de un contador con salida indicativa del contenido ($\alpha CT=9$), y entrada de modo de operación.**

A1.5.3.3 Relación de modo de operación (M)

Al igual que en los sistemas combinacionales, la relación de modo de operación M define la forma en que determinadas variables de entrada actúan sobre un bloque funcional o la forma en que éste controla las variables de salida.

En la figura A1.38 se representa, como ejemplo de modo de operación sobre una variable de entrada, un bloque funcional contador que posee una entrada de modo $M\alpha$ que cuando se encuentra en nivel uno permite que el contador realice el contaje en sentido ascendente de los impulsos que se apliquen en la entrada $\alpha+$.

En la figura A1.39 se representa un ejemplo de modo de operación que influye en una variable de salida de un contador. Cuando la entrada $M\alpha$ se encuentra en nivel uno, la salida $\alpha CT=9$ se pone a uno si el contenido de contador es 1001 (equivalente al número decimal 9).

A1.5.3.4 Combinación de relaciones de dependencia

Las relaciones de dependencia se pueden también combinar entre sí, en los bloques funcionales secuenciales. La mayor complejidad de éstos con respecto a los combinacionales, hace que dicha combinación se presente en gran cantidad de ocasiones.

a) Combinación de simultaneidad

Se especifica también mediante los símbolos adecuados separados por comas. Se presenta igualmente cuando una entrada que actúa de una sola forma sobre el contenido de un bloque funcional, está condicionada por el estado de varias entradas simultáneamente, o bien cuando el estado de una salida depende de varias entradas simultáneamente.

En la figura A1.40 se presenta un ejemplo de combinación de simultaneidad en sistemas secuenciales. La entrada inferior del bloque funcional contador lleva

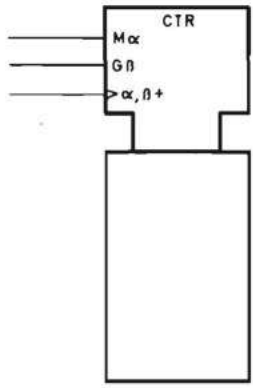


FIGURA A1.40.—**Símbolo de un contador síncrono de contaje ascendente con entrada de modo de operación ($M\alpha$) y entrada de desinhibición ($G\beta$).**

el símbolo correspondiente a la actuación por medio de flancos de subida y el indicativo +, que especifica que los impulsos aplicados a ella son contados en sentido ascendente. Pero además, en este bloque, el contaje sólo debe realizarse si la entrada de modo $M\alpha$ se encuentra en nivel uno y la entrada de inhibición $G\beta$ se encuentra también en dicho estado. Por ello la entrada a la que se aplican los impulsos de contaje recibe la denominación α, $\beta+$.

b) Combinación de exclusión

Se especifica también mediante los símbolos adecuados separados por un segmento inclinado. Se presenta cuando una entrada que actúa de varias formas sobre un bloque funcional, está condicionada por el estado de una o más entradas de forma diferente, o bien cuando una salida puede adoptar varios estados diferentes en función del estado de una o más entradas.

En la figura A1.41 se representa un ejemplo de esta forma de actuación. La entrada inferior del bloque funcional contador lleva el símbolo correspondiente a la actuación por medio de los flancos de subida seguido de los indicativos $\alpha+/\overline{\alpha}$—. Ello quiere decir que si la entrada $M\alpha$ se encuentra en nivel uno, los impulsos se cuentan en sentido ascendente y si la entrada $M\alpha$ se encuentra en nivel cero, los impulsos son contados en sentido descendente.

Pero además, en sistemas secuenciales es usual que se presenten juntas ambas formas de combinar los indicativos de dependencia. En la figura A1.42 se representa un ejemplo sencillo que constituye una combinación de los estudiados en las figuras A1.40 y A1.41. El bloque funcional contador de la figura A1.42 realiza el contaje de los flancos de subida aplicados en la entrada correspondiente de las dos formas siguientes:

— En sentido ascendente si $M\alpha$ se encuentra en nivel uno y $G\beta$ se encuentra también en dicho nivel.

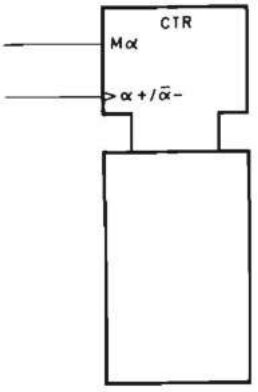


FIGURA A1.41.—**Símbolo de un contador síncrono reversible con entrada de modo de operación ($M\alpha$) que selecciona contaje ascendente y descendente.**

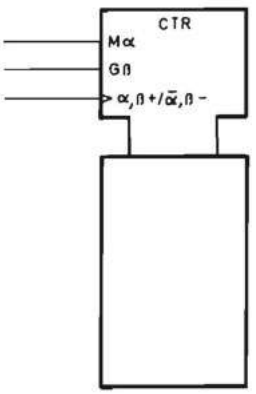


FIGURA A1.42.—**Símbolo de un contador síncrono reversible con entrada de modo de operación ($M\alpha$) que selecciona el modo de operación ascendente/descendente, y entrada de desinhibición ($G\beta$).**

— En sentido descendente si $M\alpha$ se encuentra en nivel cero y $G\beta$ se encuentra en nivel uno.

En la combinación de la simultaneidad y la exclusión, se puede aplicar la propiedad distributiva de la primera con relación a la segunda para simplificar así los símbolos normalizados. Como ejemplo, se puede aplicar a la figura A1.42, obteniendo la versión simplificada de la figura A1.43.

Las combinaciones de simultaneidad y exclusión se pueden aplicar también a las variables de salida. En la figura A1.44 se representa como ejemplo un contador con una entrada de modo de operación $M\alpha$ y una salida de indicación de contenido que tiene asignados los símbolos $\overline{\alpha CT=0}/\alpha CT=15$, que especifican que el

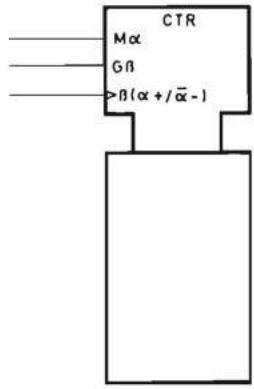


FIGURA A1.43.—**Símbolo equivalente al de la figura A1.42.**

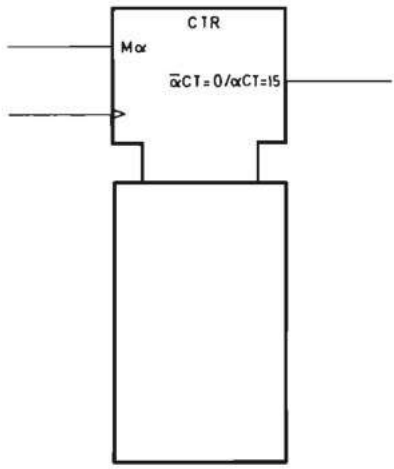


FIGURA A1.44.—**Símbolo de un contador síncrono con entrada de modo de operación ($M\alpha$) que selecciona la forma de operación de la salida indicativa del contenido ($\bar{\alpha}CT=0/\alpha CT=15$).**

terminal correspondiente se pone a uno si se produce cualquiera de las dos situaciones siguientes:

— La entrada $M\alpha$ se encuentra en nivel cero y el contenido del contador es equivalente al número decimal cero.
— La entrada $M\alpha$ se encuentra en nivel uno y el contenido del contador es equivalente al número decimal quince.

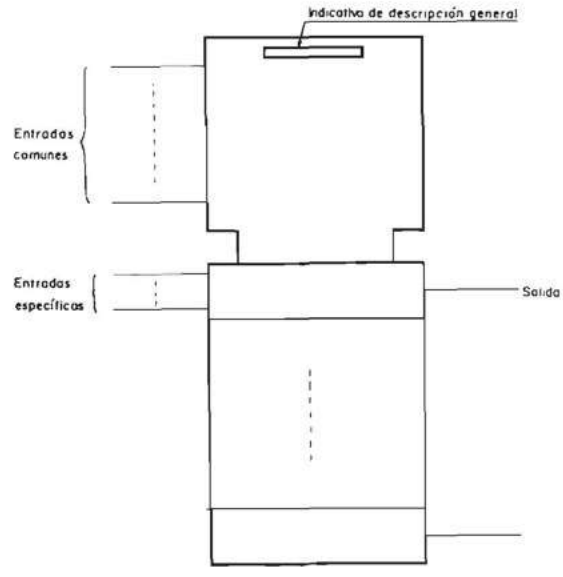


FIGURA A1.45.—**Símbolo normalizado general de un bloque funcional secuencial.**

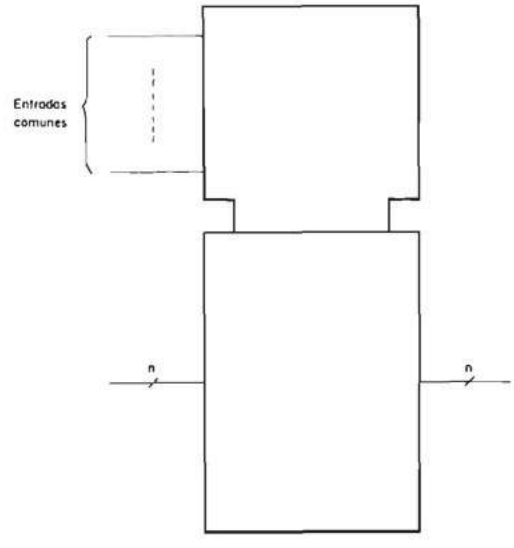


FIGURA A1.46.—**Símbolo normalizado general de los registros de entrada y salida en paralelo.**

A1.5.4 Ejemplos prácticos de sistemas secuenciales

En este apartado se analizan diferentes bloques funcionales secuenciales para comprobar que las normas de representación estudiadas permiten prescindir de las tablas de verdad o tablas de funcionamiento y obtener, de esta forma, esquemas autoexplicativos.

Los diferentes bloques funcionales secuenciales presentan la característica general común de poseer varios biestables y en su caso un circuito combinacional. Ambos se caracterizan por poseer entradas específicas y entradas comunes y, por ello, el símbolo general es el representado en la figura A1.45.

En sucesivos apartados se analizan distintos tipos de sistemas secuenciales.

A1.5.4.1 Registros de entrada y salida en paralelo

Constituye el bloque funcional síncrono más sencillo cuya misión es la de almacenar información.

En la figura A1.46 se representa el símbolo general de los registros de entrada y salida en paralelo, que se caracteriza por:

a) Carecer de indicativo de descripción general.
b) Poseer *n* entradas y *n* salidas de información.
c) Poseer entradas comunes de control, de las que las más usuales son:

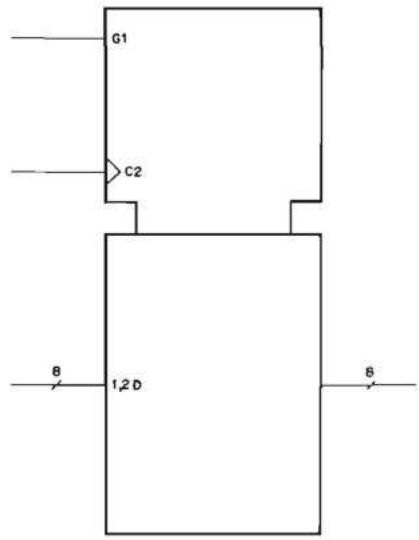


FIGURA A1.47.—**Símbolo normalizado de un registro de entrada y salida en paralelo de 8 bits con terminal de desinhibición de la entrada de información (*G1*).**

— Control de entrada de información que actúa por flanco o por nivel (*C*).
— Desinhibición de la entrada de información (*G*).
— Desinhibición de la salida (*EN*).
— Puesta a cero asíncrona (*R*).

En la figura A1.47 se representa un registro de entrada y salida en paralelo de 8 bits, formado por biestables *D*, que posee un terminal de desinhibición de la entrada de información (Clock enable) (*G*1) y un terminal de impulsos de entrada de información (*C*2). La información presente en las entradas de los 8 biestables se almacena en su interior cuando *G*1 se encuentra en nivel uno y se aplica un flanco de subida en *C*2, lo cual se indica mediante el conjunto de indicativos «1,2 *D*» colocados en los terminales de entrada.

En la figura A1.48 se representa un registro de entrada y salida en paralelo de 4 bits, con un terminal de puesta a cero asíncrona (*R*), un terminal de desinhibición del tercer estado de la salida (*EN*) y un terminal de impulsos de sincronismo o entrada de información síncrona (*C*1). La información presente en las entradas se memoriza en el registro cuando se aplica un flanco de subida en el terminal *C*1, siempre y cuando *R* no se encuentre en nivel uno. Para indicarlo se coloca en la entrada en paralelo del primer biestable el indicativo 1*D*.

Los registros de entrada y salida en paralelo se combinan con diferentes tipos de bloques funcionales combinacionales como son los decodificadores y los multiplexores.

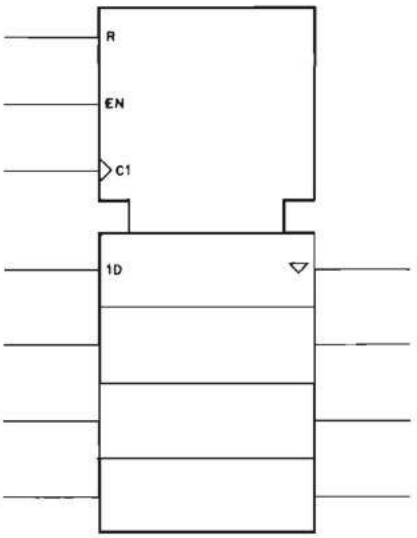


FIGURA A1.48.—**Símbolo normalizado de un registro de entrada y salida en paralelo de 4 bits con terminales de puesta a cero asíncrona y de desinhibición de la salida.**

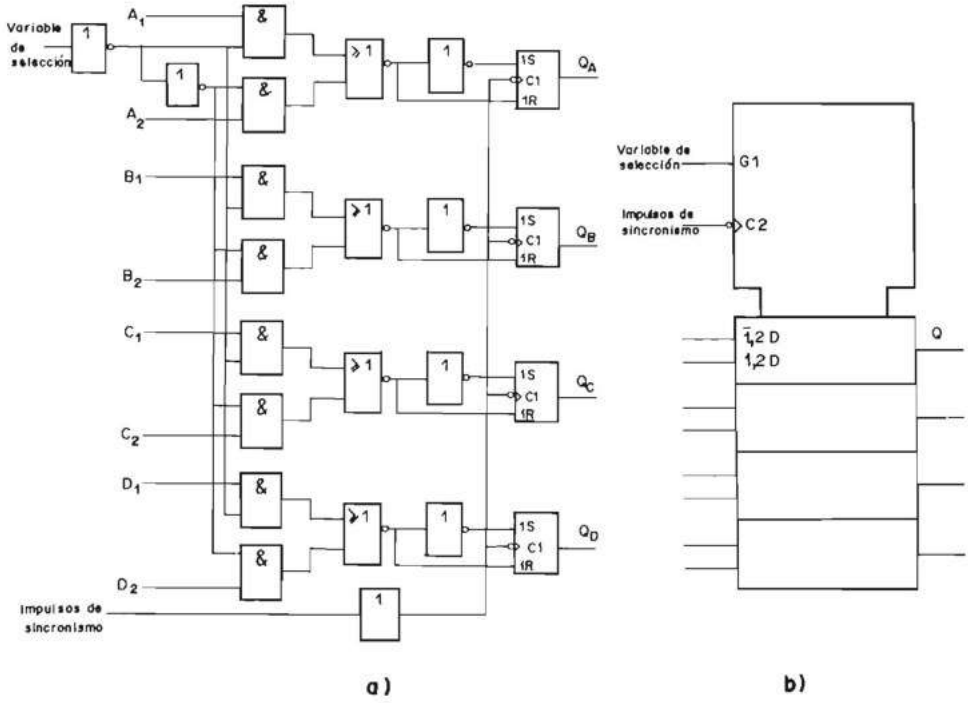


FIGURA A1.49.—*a*) **Esquema de un registro de entrada y salida paralelo de 4 bits con un cuádruple multiplexor de dos canales a su entrada;** *b*) **Símbolo normalizado.**

En la figura A1.49*a* se representa un circuito formado por un registro de 4 bits de entrada y salida en paralelo a cuyas entradas se conecta un cuádruple multiplexor de dos canales de entrada que posee una variable de selección. Según esta variable se encuentre en nivel cero o uno, en las entradas del registro aparece la información de los terminales A_1 a D_1 o A_2 a D_2 respectivamente.

En la figura A1.49*b* se representa el símbolo normalizado correspondiente. El terminal *G*1 corresponde a la variable de selección y *C*2 a la entrada de impulsos de sincronismo que producen la memorización en los flancos de bajada. Cada elemento del bloque posee dos terminales de entrada que llevan asociados los indicativos «$\overline{1}$,2*D*» y «1,2*D*» respectivamente.

Los indicativos «$\overline{1}$,2*D*» denotan que la información presente en la entrada correspondiente se memoriza en el registro cuando *G*1 se encuentra en nivel cero ($\overline{1}$) y se aplica un flanco activo en *C*2 (2*D*). Los indicativos «1,2*D*» indican que la información presente en la entrada correspondiente se memoriza en el registro cuando *G*1 se encuentra en nivel uno (1) y se aplica un flanco activo en *C*2 (2*D*).

En la figura A1.50*a* se representa un multiplexor de ocho canales que posee un registro de las variables de selección y otro de las variables de entrada, que tienen variables de control independientes *C*8 y *C*9, activa con nivel cero la primera y con flancos de subida la segunda. En la figura A1.50*b* se representa el símbolo gráfico

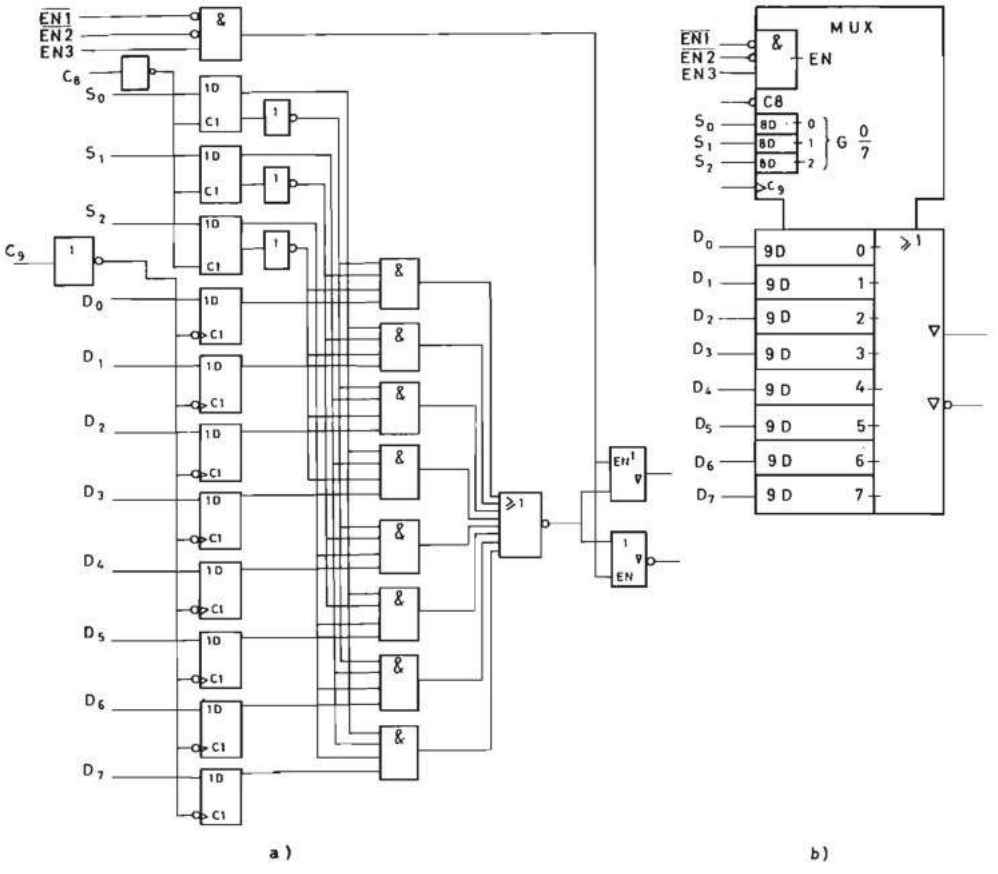


FIGURA A1.50.—*a)* **Esquema de un multiplexor de ocho canales con registro de las variables de selección y registro de las variables de entrada; *b)* Símbolo normalizado.**

normalizado en cuya parte común se colocan las entradas de control *C*8 y *C*9, las de selección $G\ \frac{0}{7}$ y la de desinhibición de la salida de tres estados *EN* que se describen seguidamente.

La desinhibición de la salida *EN* está constituida por la función lógica *Y* (&) de las tres entradas $\overline{EN}1$, $\overline{EN}2$ y *EN*3.

Las variables de selección S_0, S_1 y S_2 se introducen en un registro de tres bits cuando la entrada *C*8 se encuentra en nivel cero y quedan memorizadas cuando ésta pasa el nivel uno. La memorización se indica colocando el indicativo 8*D* en las entradas de selección y el nivel cero activo de *C*8 mediante el símbolo de inversión.

Las entradas de información D_0 a D_7 se introducen y quedan memorizadas en un registro cuando se aplica un flanco de cero a uno en *C*9. Dicha acción se indica colocando los indicativos 9*D* en las entradas correspondientes. Las salidas de este

registro se seleccionan para presentarse a las 8 entradas de la puerta O mediante las salidas del registro de selección que actúan a través de una relación de dependencia *Y(G)*.

Por ello, en estas últimas se coloca el indicativo $G\ \frac{0}{7}$ y en las primeras los números de 0 a 7. Si las variables $G\ \frac{0}{7}$ se encuentran en los niveles $101_2 \equiv 5_{10}$, se selecciona la salida del biestable 5 para aplicarse a la entrada de la puerta O.

A1.5.4.2 Contadores

A1.5.4.2.1 Contadores asíncronos

El símbolo general se representa en la figura A1.51. Se caracteriza por el hecho de que la entrada de contaje no forma parte de las entradas comunes y está constituida por la entrada de control *C* de uno de los biestables.

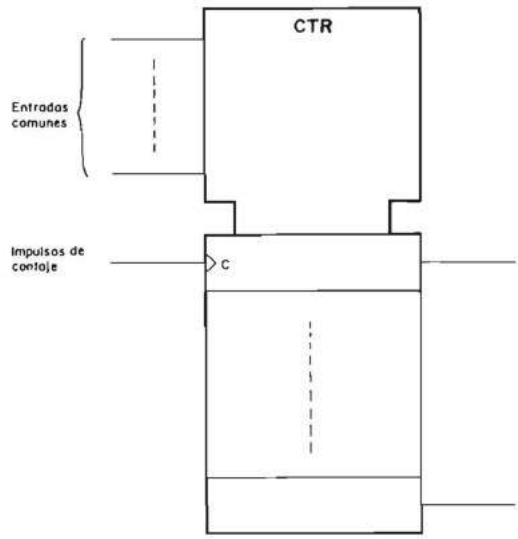


FIGURA A1.51.—**Símbolo normalizado general de un contador asíncrono.**

Veamos primeramente la forma de obtener el símbolo lógico a partir de las tablas de funcionamiento. Se supone un contador asíncrono en BCD natural con entrada de puesta a cero asíncrona e inhibición de contaje, que realiza el contaje de los flancos de bajada de la señal de entrada.

En la tabla A1.5 se indica la forma en que actúan las diferentes entradas y en la tabla A1.6 el código de contaje.

R	$G1$	$1C+$	Q_3	Q_2	Q_1	Q_0
1	X	X	0	0	0	0
0	0	X	Inhibición			
0	1	↓	Contaje			

TABLA A1.5

Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

TABLA A1.6

El símbolo normalizado de la figura A1.52 contiene toda la información de las tablas A1.5 y A1.6 y las hace innecesarias. El indicativo general CRTDIV10 indica que se trata de un contador que divide por diez, es decir, cuenta de 0 a 9 (la ausencia de indicación del código de contaje indica que se trata del BCD natural). La colocación de la entrada de contaje en el primer biestable indica que se trata de un contador asíncrono. El indicativo $1C+$ especifica que $G1$ ha de estar en nivel uno (1) para que se realice el contaje, y que éste se efectúa en sentido ascendente (+).

En la figura A1.53 se presenta como ejemplo el símbolo lógico del circuito integrado contador asíncrono de 4 bits en binario natural (4 bits binary ripple counter) 7493. Este circuito posee un biestable que constituye un divisor por dos y tres biestables más que forman un divisor por ocho. El divisor por dos posee una entrada de contaje que se activa con el flanco de bajada de los impulsos de contaje, y lo mismo le sucede al divisor por ocho. El contador posee dos entradas que, cuando se encuentran ambas en nivel uno (función &), realizan la puesta a cero (R) de todos los biestables.

A1.5.4.2.2 Contadores síncronos

El símbolo general de los contadores síncronos se representa en la figura A1.54, que se diferencia de la figura A1.51 en que el terminal de los impulsos

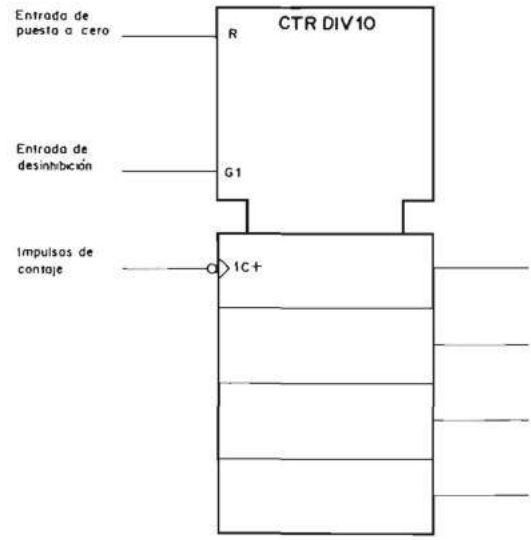


FIGURA A1.52.—**Símbolo normalizado de un contador asíncrono en BCD natural con entrada asíncrona de puesta a cero (*R*) y entrada de desinhibición del contaje (*G*1).**

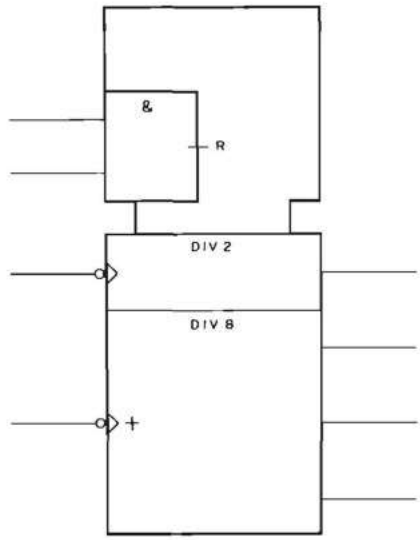


FIGURA A1.53.—**Símbolo normalizado del circuito contador asíncrono 7493.**

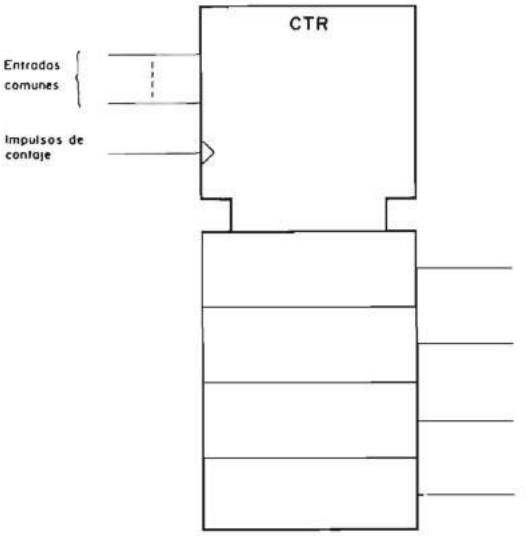


FIGURA A1.54.—**Símbolo normalizado general de un contador síncrono.**

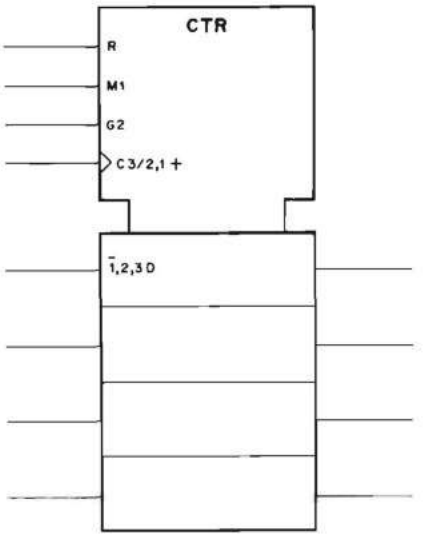


FIGURA A1.55.—**Símbolo normalizado de un contador síncrono con entrada de puesta a cero asíncrona (*R*), entrada de selección de modo de operación (*M*1) (contaje/paralelo) y entrada de desinhibición del contaje y de la entrada en paralelo (G2).**

de contaje está situado en la parte superior, porque es común a todos los biestables del contador.

Estos presentan una gran cantidad de opciones en las entradas comunes de acuerdo con las funciones que incorpora el circuito. En la figura A1.55 se representa el símbolo gráfico normalizado de un contador síncrono que posee:

a) Una entrada de puesta a cero asíncrona *R*.
b) Una entrada de modo de operación *M*1 que permite seleccionar entre el contaje en sentido ascendente, especificado mediante los indicativos 1 + en la entrada de impulsos de sincronismo, y la entrada en paralelo, especificada mediante el indicativo $\overline{1}$ en la entrada de cada biestable.
c) Una entrada de desinhibición *G*2 que debe encontrarse en nivel uno para que se desinhiban tanto la entrada en paralelo como el contaje.
d) Una entrada de impulsos de sincronismo que realiza tanto la entrada en paralelo (especificada mediante los indicativos *C*3) como el contaje (especificado mediante los indicativos 2, 1 + que indican, que simultáneamente deben encontrarse en nivel uno los terminales *G*2 y *M*1).
Dado que el contaje y la entrada en paralelo son acciones alternativas, esta entrada recibe la denominación «*C*3/2,1 + ».
e) Una entrada de información en paralelo especificada mediante los indicativos $\overline{1}$,2,3*D* que nos muestran que para que se realice esa acción es necesario que simultáneamente se encuentre *M*1 en nivel cero ($\overline{1}$), *G*2 en nivel uno (2) y que se aplique un flanco activo en *C*3 (3*D*).

Un ejemplo práctico es el circuito integrado 74191 que constituye un contador síncrono reversible en binario natural de 4 bits con entradas en paralelo asíncronas, desinhibición de contaje para ampliar la capacidad de contaje de forma asíncrona y entrada de selección de contaje ascendente o descendente.

Entradas					Salidas
$\overline{Ep}$ (C4)	$\overline{A}/D$ (*M*2)	$\overline{E}_A$ (G1)	*T*(1,$\overline{2}$—/1,2+) (*G*3)	(4*D*)	Q_{n-}
0	X	X	X	0	0
0	X	X	X	1	1
1	0	0	↑	X	Contaje ascendente
1	1	0	↑	X	Contaje descendente
1	X	1	X	X	Inhibición del contaje

TABLA A1.7

En la tabla A1.7 se presenta la tabla de funcionamiento y en la A1.8, la tabla de verdad de las salidas de propagación del contaje. Ambas tablas quedan perfectamente especificadas en el símbolo gráfico normalizado de la figura A1.56, en la que la función de los terminales de entrada y salida es la siguiente:

Entradas			Contenido				Salidas	
$\overline{A}/D$ (M2)	$\overline{E}_A$ (G1)	$T(1,\overline{2}-/1,2+)$ (G3)	Q_3	Q_2	Q_1	Q_0	Max/Min	$\overline{P}_1$
1	1	X	1	1	1	1	0	1
0	1	X	1	1	1	1	1	1
0	0	⊓	1	1	1	1	⌐	⊓
0	1	X	0	0	0	0	0	1
1	1	⊓	0	0	0	0	1	1
1	0		0	0	0	0	⌐	⊓

TABLA A1.8

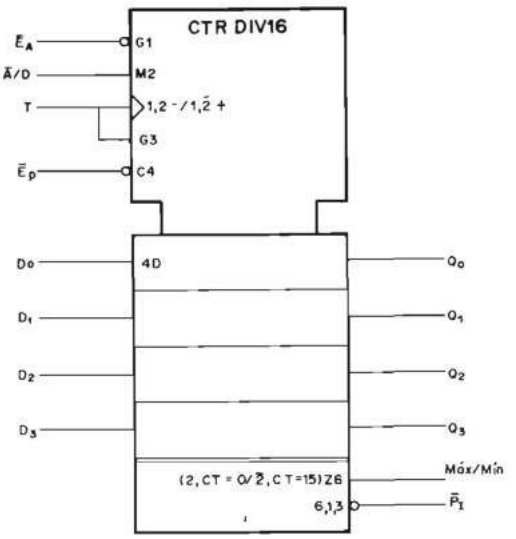


FIGURA A1.56.—**Símbolo normalizado del contador síncrono 74191.**

$\overline{E}_A$: Entrada de propagación de acarreo.
$\overline{A}/D$: Entrada de modo de contaje ascendente o descendente.
T: Entrada con una doble función:
— Contaje: Ascendente si $G1 = 1$ y $M2 = 0$
Descendente si $G1 = 1$ y $M2 = 1$
— Inhibición de la propagación.
$\overline{E}p$: Entrada de control de la entrada de información en paralelo.
Dn (4D): Entrada de información en paralelo. El valor lógico presente en ella se introduce en el biestable correspondiente cuando $C4 = 0$.

Max/Min [2,$CT=0/\overline{2}$, $CT=15$)Z6]: Salida que indica que el contenido del contador ha alcanzado el máximo de su capacidad de contaje. Esta salida se pone a uno si $M2=1$ (2) y el contenido del contador es cero, o bien si $M2=0$ (2) y el contenido del contador es quince. Esta salida está unida a la función $\overline{P}_1$.

$\overline{P}_1$ (6,1,3): Salida que adopta el nivel cero si Max/Min = 1, $G1=1$ y $G3=1$.

A1.5.4.3 Registros de desplazamiento

Constituyen un bloque funcional síncrono en el que aparecen múltiples opciones.

En la figura A1.57 se representa el símbolo general de un registro de desplazamiento que posee una entrada serie y una salida serie.

En la figura A1.58*a* se representa un ejemplo constituido por el circuito integrado 7491 que es un registro de desplazamiento de 8 bits (SRG8) que posee:

a) Una entrada de impulsos de desplazamiento hacia la derecha activa con flancos de subida ($C1\rightarrow$).
b) Una entrada de información serie formada por la función Y de dos terminales de entrada.
c) Una salida de información serie y su inversa.

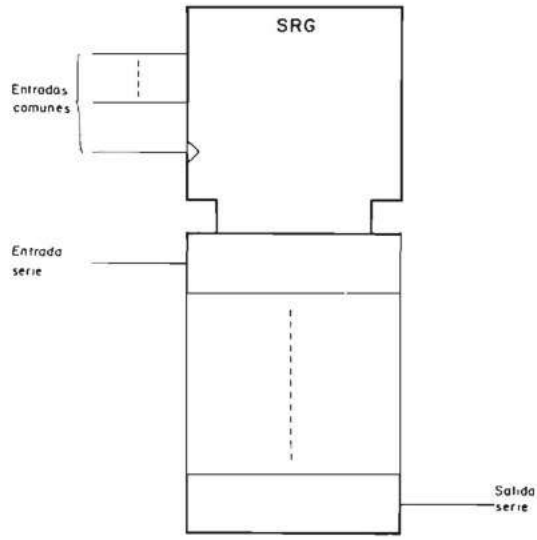


FIGURA A1.57.—**Símbolo normalizado general de un registro de desplazamiento de entrada y salida serie.**

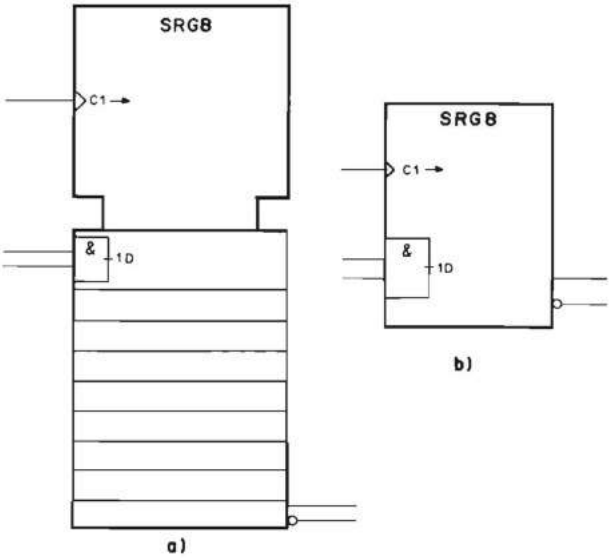


FIGURA A1.58.—**Símbolo normalizado del registro de desplazamiento 7491.**

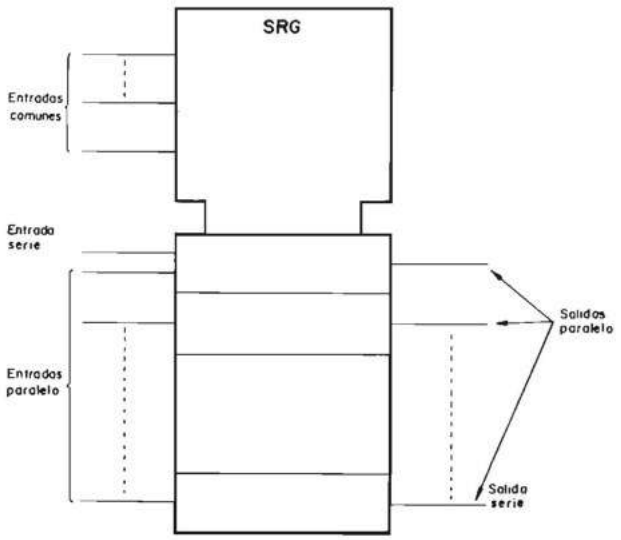


FIGURA A1.59.—**Símbolo normalizado de un registro de desplazamiento de entradas y salidas en serie y en paralelo.**

El símbolo de la figura A1.58*a* se puede simplificar, obteniéndose el de la figura A1.58*b*. Dicha simplificación es posible porque SRG8 indica que se trata de un registro de desplazamiento de 8 bits.

Pero los registros de desplazamiento se utilizan especialmente para transmitir en serie una información generada en paralelo o bien para realizar la operación inversa, es decir, obtener en paralelo una información transmitida en serie. En la figura A1.59 se representa el símbolo normalizado general de un registro de desplazamiento que puede realizar ambas funciones y que, por ello, posee entrada serie y entradas en paralelo, salida serie y salidas en paralelo y entradas de control comunes.

Un ejemplo práctico lo constituye el circuito integrado 74LS194, que constituye un registro de desplazamiento bidireccional de 4 bits, cuyo símbolo normalizado se representa en la figura A1.60 y cuyos terminales tienen las siguientes funciones:

R: Puesta a cero asíncrona de los biestables del registro cuando adopta el nivel cero.

S_0, S_1 ($M\frac{0}{3}$): Entradas de selección del modo de operación

*M*0: Inhibición del desplazamiento.
*M*1: Desplazamiento hacia la derecha.
*M*2: Desplazamiento hacia la izquierda.
*M*3: Entrada en paralelo.

*C*4/1→/2←: Entrada de impulsos activa con flancos de subida cuya acción depende del estado de las entradas de selección de modo de operación $M\frac{0}{3}$.

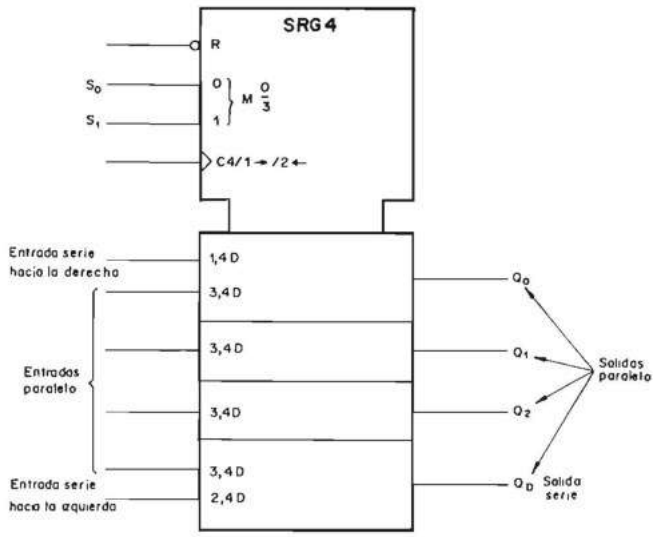


FIGURA A1.60.—**Símbolo normalizado del registro de desplazamiento 74LS194.**

Si $M\,\frac{0}{3} = 0$, los impulsos aplicados no realizan acción alguna.

Si $M\,\frac{0}{3} = 1$, los impulsos desplazan la información en serie hacia la derecha ($1\rightarrow$).

Si $M\,\frac{0}{3} = 2$, los impulsos desplazan la información hacia la izquierda ($2\leftarrow$).

Si $M\,\frac{0}{3} = 3$, se produce la entrada de información en paralelo (C4).

1,4*D*: Entrada de información en serie hacia la derecha. Para que esta acción se produzca, en las entradas S_1 y S_0 debe aplicarse la combinación 01, equivalente a $1_{10}\left(M\,\frac{0}{3} = 1\right)$.

2,4*D*: Entrada de información en serie hacia la izquierda. Para que esta acción se produzca, en las entradas S_1 y S_0 debe aplicarse la combinación 10, equivalente a $2_{10}\left(M\,\frac{0}{3} = 2\right)$.

3,4*D*: Entradas de información en paralelo. Para que esta acción se produzca, en las entradas S_1 y S_0 debe aplicarse la combinación 11, equivalente a $3_{10}\left(M\,\frac{0}{3} = 3\right)$.

Apéndice **2**

Circuitos temporales analógico-digitales

A2.1 INTRODUCCION

La gran cantidad de aplicaciones de los circuitos temporales analógico-digitales impulsó que diversos fabricantes los realizasen en circuito integrado.

Analizaremos en este apéndice algunos de los circuitos disponibles comercialmente. Para un estudio más detallado se remite al lector a la bibliografía [INTE 82] [MAXI 82] [NATI 82] [SIGN 85].

A2.2 CIRCUITO TEMPORAL MULTIFUNCIONAL 555

El circuito 555 responde al esquema representado en la figura 6.111 y se encapsula en formato normalizado de doble línea (DIL) de 8 terminales. Su diagrama de bloques se representa en la figura A2.1. El biestable utilizado en el 555 es de activación por niveles.

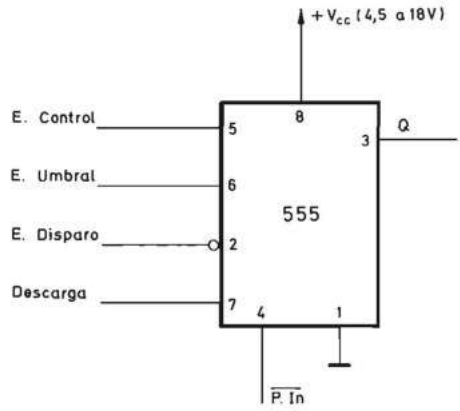


FIGURA A2.1.—**Circuito temporal analógico-digital 555.**

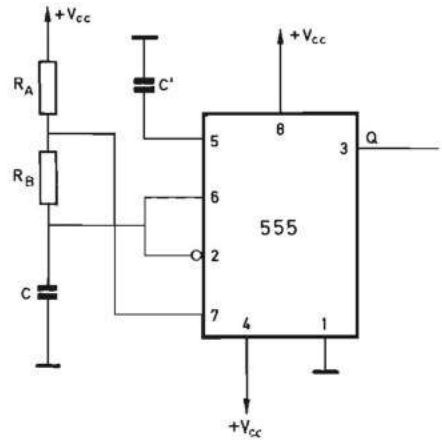


FIGURA A2.2.—**Generador de impulsos realizado con el circuito 555.**

En la figura A2.2 se representa el esquema de un astable o generador de impulsos realizado con el 555 que es equivalente al esquema de la figura 6.109.

En la figura A2.3 se representa el circuito de un monoestable no redisparable activo con el nivel bajo de la señal de disparo (nivel cero en lógica positiva). Se puede comprobar la equivalencia entre este montaje y el de la figura 6.110. El terminal 5 se podría dejar al aire pero la conexión de un condensador entre él y masa (valor típico 0,1 μF) ayuda a eliminar el rizado o el ruido que pueda aparecer en la fuente de alimentación. La duración del impulso a la salida es aproximadamente igual a 1,1 *RC*.

El monoestable de la figura A2.3 constituye un temporizador no redisparable porque, tal como se indica en la figura A2.4, si durante el intervalo *t* de temporiza-

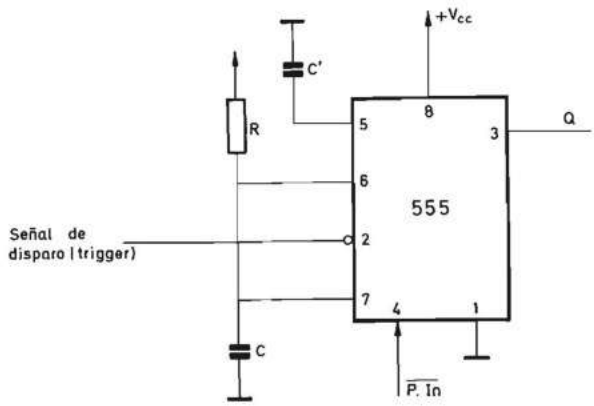


FIGURA A2.3.—**Monoestable no redisparable realizado con el circuito 555.**

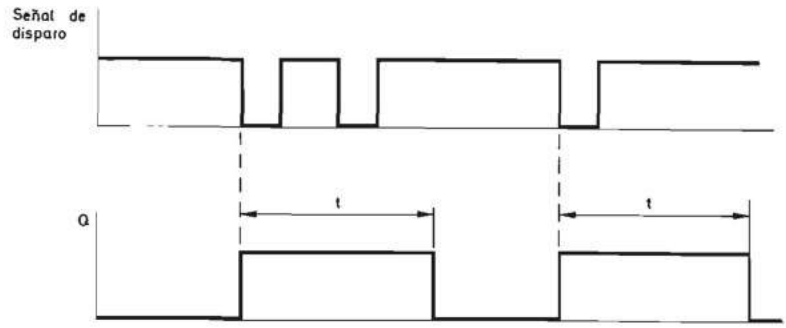


FIGURA A2.4.—**Diagrama de evolución en el tiempo de las señales del monoestable de la figura A2.3.**

ción se aplica un nuevo impulso de disparo, el citado intervalo no varía. Se puede realizar un monoestable redisparable con un temporizador analógico-digital si el impulso de disparo, además de actuar sobre el estado del biestable, produce la descarga del condensador. En la figura A2.5 se representa el esquema de un monoestable redisparable realizado con el monoestable 555 que se diferencia del esquema de la figura A2.3 por la presencia del transistor *T* (del tipo *PNP*), cuyo emisor está unido a la entrada de descarga y su base a la señal de disparo. Al hacer bajar ésta a nivel cero durante el intervalo de temporización se produce la descarga del condensador a través del transistor *T* y, por tanto, se vuelve a iniciar dicho intervalo. En la figura A2.6 se representa el cronograma de este circuito, cuyo comportamiento puede ser estudiado desde el punto de vista de la frecuencia de los impulsos

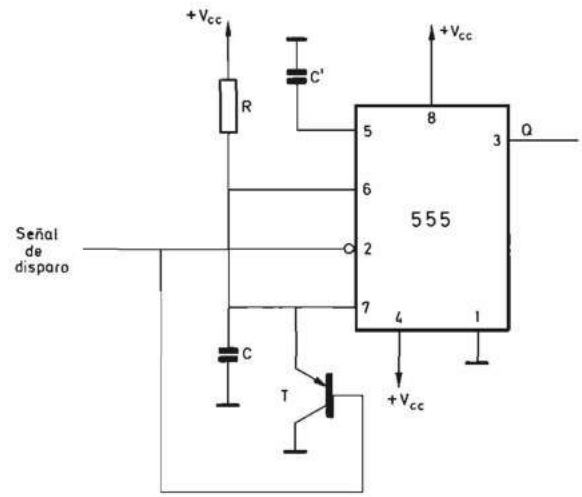


FIGURA A2.5.—**Monoestable redisparable realizado con el circuito 555.**

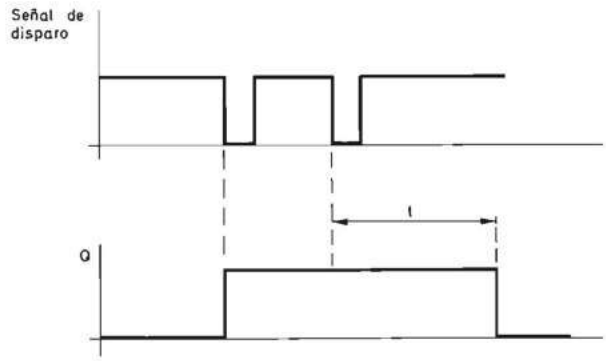


FIGURA A2.6.—**Diagrama de evolución en el tiempo de las señales del monoestable de la figura A2.5.**

de disparo. Si dicha frecuencia es superior a $1/t$, la salida Q permanece siempre en nivel alto. Por tanto, este circuito puede ser utilizado como detector de falta de impulsos de frecuencia superior a $1/t$.

La disponibilidad al exterior de la entrada inversora (—) del amplificador operacional A (entrada de control) de la figura 6.110 permite la realización de circuitos con numerosas aplicaciones.

Las ventajas de la transmisión de información en un dominio temporal hace muy interesante la conversión de variables analógicas en temporales. Un convertidor de este tipo se puede realizar mediante un temporizador analógico-digital. En la figura A2.7 se representa el esquema de un convertidor de la amplitud de una señal analógica en la relación alto-bajo de una secuencia de impulsos.

La señal analógica se aplica a la entrada de control (5) y una onda cuadrada de frecuencia adecuada a la entrada de disparo (2). En la salida Q_T aparecen im-

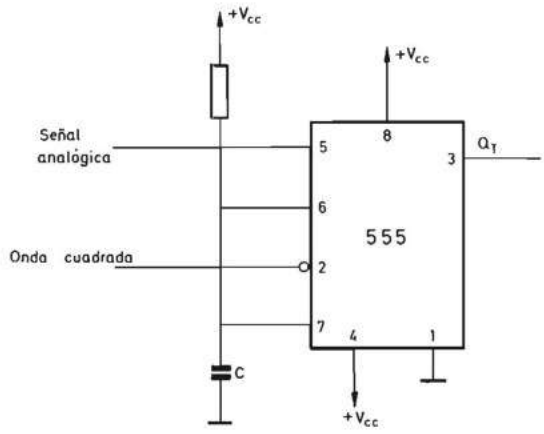


FIGURA A2.7.—**Circuito de un convertidor de amplitud de una señal analógica en anchura de impulso.**

pulsos cuya anchura (permanencia en nivel uno) es proporcional al nivel de la señal analógica que ha de permanecer constante durante un ciclo del generador y conectarse a 5 a través de una baja impedancia.

Se pueden acoplar entre sí un número ilimitado de circuitos 555 conectados según el esquema de la figura A2.2, para obtener un temporizador que genera una secuencia de impulsos cuya duración es programable individualmente. En la figura A2.8*a* se representa el sistema constituido por tres circuitos 555, y en la figura A2.8*b* la secuencia de las señales obtenidas. La señal de disparo se aplica a la entrada correspondiente (terminal 2) del primer circuito 555 en el que el producto *RC* fija la duración del impulso a su salida Q_1. La finalización del impulso en la salida Q_1 debe activar al segundo 555 para que se inicie la generación del impulso

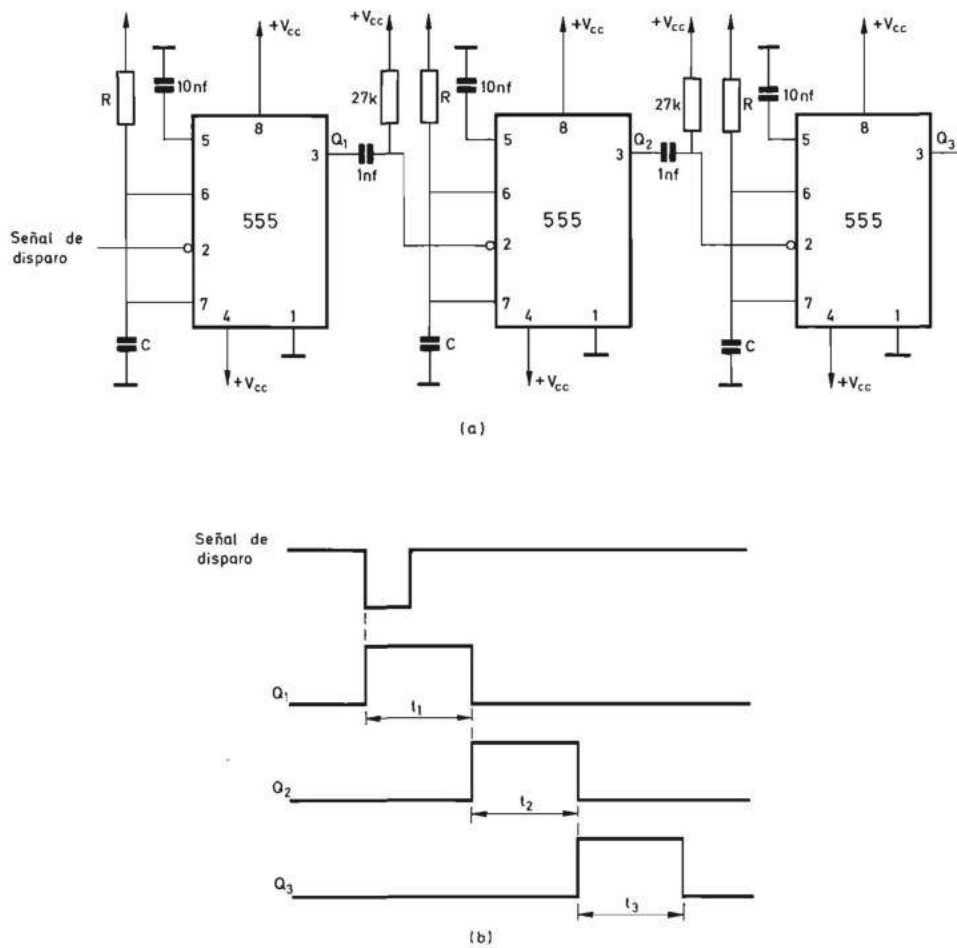


FIGURA A2.8.—**Temporizador que genera una secuencia de impulsos, realizado mediante el acoplamiento de circuitos 555, y la secuencia de señales generada por él.**

en Q_2. Dado que el biestable que forma parte del 555 se activa con el nivel cero de la entrada de disparo, no es posible la unión directa de la salida Q_1 del primer 555 con la entrada de disparo (terminal 2) del segundo 555 porque éste estaría activado de forma continuada. Por ello dicha conexión se realiza a través de un circuito derivador constituido por un condensador C_1 de 1 nF y una resistencia R_1 de 27 kΩ. La bajada a cero de Q_1 hace que el punto común a C_1 y R_1 pase a nivel cero durante un pequeño intervalo de tiempo suficiente para disparar el segundo circuito 555. El segundo temporizador se conecta al tercero a través de un condensador C_2 y una resistencia R_2 idénticos a C_1 y R_1 respectivamente. Las redes RC temporizadoras de los tres circuitos 555 de la figura A2.8 son idénticas y por ello $t_1 = t_2 = t_3$. Seleccionando adecuadamente dichas redes se pueden obtener impulsos de diferente duración.

A2.3 CIRCUITO TEMPORAL MULTIFUNCIONAL 556

El circuito 556 contiene simplemente dos bloques idénticos al 555. Por ello permite reducir la complejidad de aquellos sistemas electrónicos que utilizan dos o más bloques funcionales idénticos al 555.

En la figura A2.9 se representa un temporizador de dos etapas realizado con un único circuito 556, cuyo comportamiento es idéntico al de las dos primeras etapas del circuito de la figura A2.8.

A2.4 CIRCUITO TEMPORAL ANALOGICO-DIGITAL 558

El circuito 558 posee cuatro temporizadores cuyo diagrama de bloques se repre-

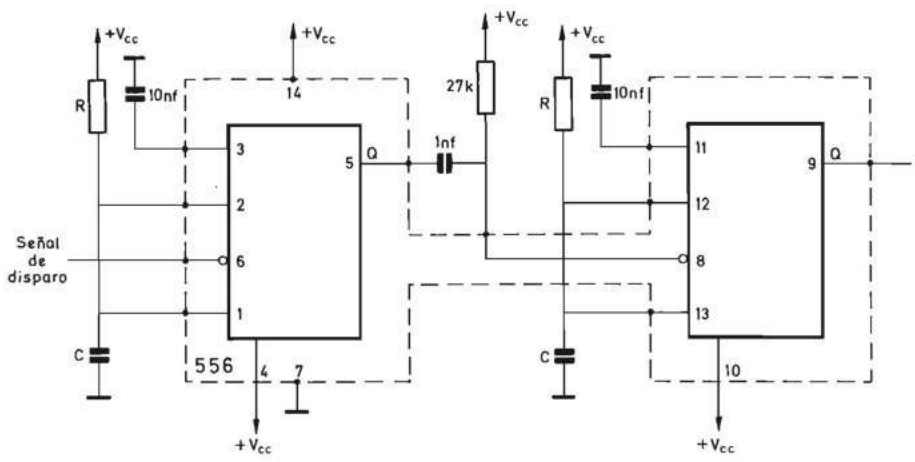


FIGURA A2.9.—**Temporizador de dos etapas realizado mediante el circuito integrado 556.**

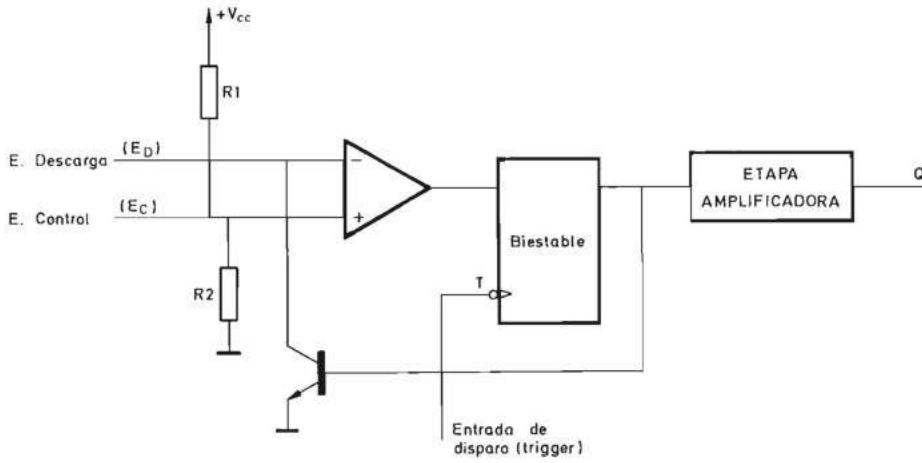


FIGURA A2.10.—**Esquema del temporizador integrado en el circuito 558.**

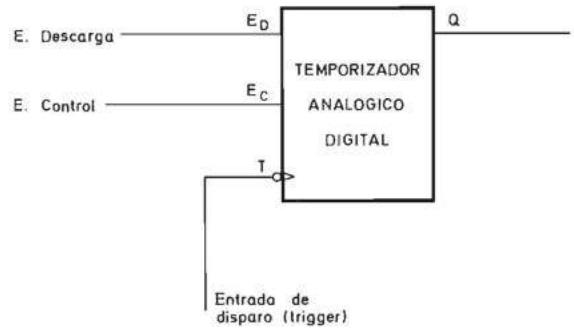


FIGURA A2.11.—**Diagrama de bloques del temporizador de la figura A2.10**

senta en la figura A2.10 que corresponde al de la figura 6.92 analizado en el apartado 6.2.4.2.2. Las entradas inversora (—) y no inversora (+) del comparador se disponen al exterior. La primera recibe el nombre de entrada de descarga (discharge input) o entrada de temporización (timing input), y la segunda, entrada de control (control input). Las resistencias R_1 y R_2 son comunes a los cuatro temporizadores contenidos en el 558, cuya entrada de disparo (trigger input) es activa con los flancos de bajada. El circuito de la figura A2.10 se puede representar mediante el diagrama de bloques de la figura A2.11.

En la figura A2.12 se representa un temporizador de cuatro etapas realizado con un único circuito 558, equivalente a los de las figuras A2.8 y A2.9. Al ser la entrada de disparo de cada uno de los temporizadores activa con los flancos de bajada,

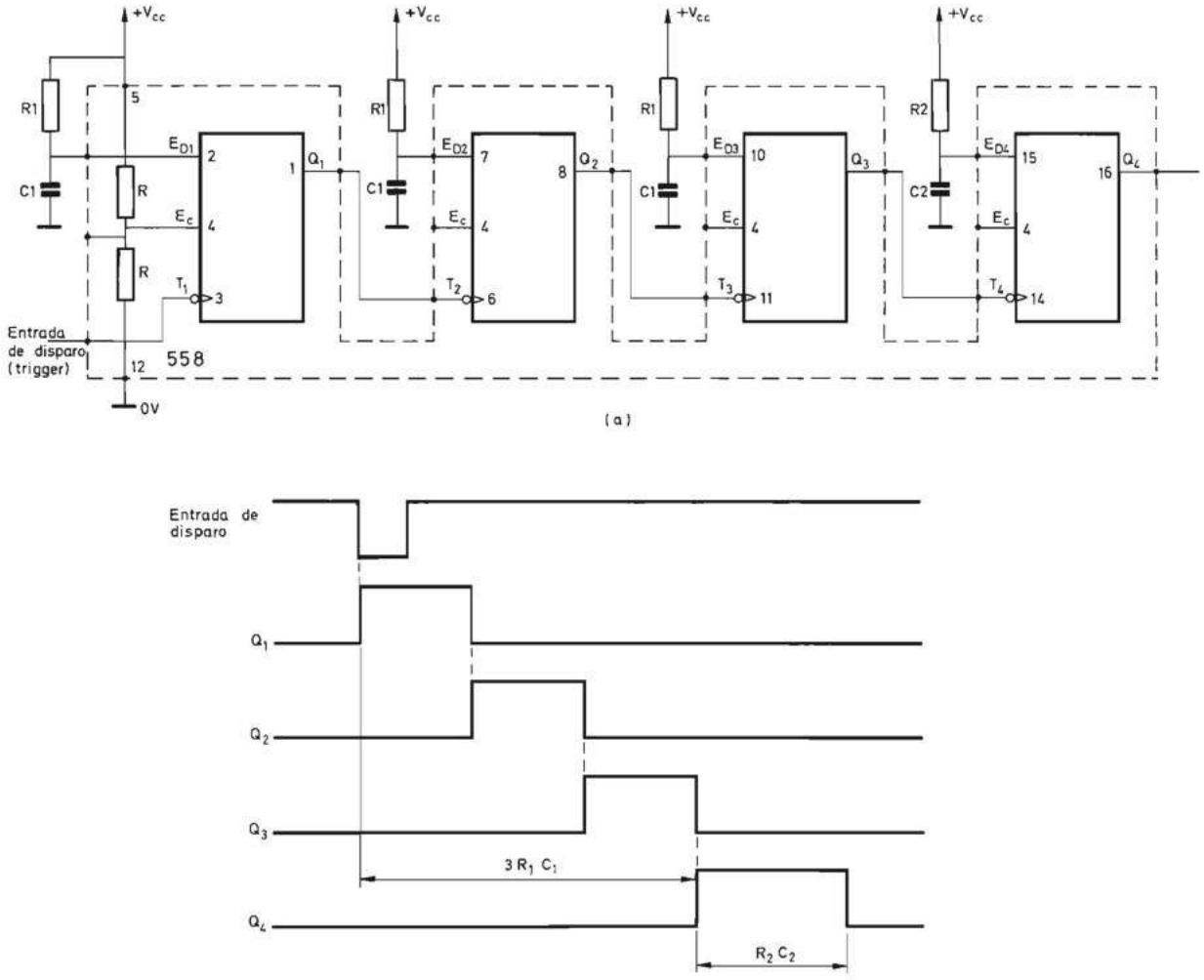


FIGURA A2.12.—**Temporizador de cuatro etapas realizado con un único circuito integrado 558 y diagrama de evolución en el tiempo de sus señales.**

se puede conectar la salida Q de cada uno de ellos a la entrada T del siguiente sin necesidad de intercalar entre ellas la red RC necesaria en el sistema de las figuras A2.8 y A2.9.

BIBLIOGRAFIA

[INTE 82] Integrated circuits databook. pp. 6.143 a 6.148. 1982.
[MAXI 85] CMOS. Data Adquisition Catalog. 1985.
[NATI 82] National Semiconductor Corporation. Linear Databook. pp. 9.33 a 9.42. 1982.
[SIGN 85] Linear LSI handbook. I C11N. New series. 1985.

Indice alfabético

Absorción, ley de, 24.
Acarreo, 181:
  en paralelo, 184.
  en serie, 183.
  generador de, 185.
Acceso:
  aleatorio, 144, 581.
  directo, memoria, 579.
  forma de, 578.
  secuencial, 579.
  serie, 636.
  tiempo de, 582.
ADLP, 255.
Adyacencia, 10.
Aiken, código, 8.
Aleatorio:
  acceso, 144, 581.
  fenómeno estático, 80.
  fenómeno dinámico, 86.
  fenómeno aleatorio en sistemas secuenciales asíncronos, 368.
Alfanumérico, código, 13.
Algebra de Boole:
  definición, 2.
  postulado, 21.
  teoremas, 22.
APLD, 255.
Aritmética binaria, 179.
Arseniuro de galio, 311.
ASCII, código, 13.
ASIC, 250, 261.
Asíncrono, sistema secuencial, 318, 320.
  caracterización por flancos, 318, 341.
  caracterización por niveles, 318, 321.
Asociativas, memorias, 659.
ASSP, 250.
Astable, circuito, 385.
Autómata finito:
  de Mealy, 316.
  de Moore, 316.
Barra de información (bus), 512, 633.
Base de numeración, 1.
BCD, códigos:
  Aiken, 8.
  exceso tres, 9.
  natural, 8.
BICMOS, 309.
Biestables:
  asíncronos activados por flancos, 354.
  asíncronos activados por niveles, 331.
  CAF, 356.
  sincronizados, 404:
    cerrojo (latch), 407.
    D, 412, 421.
    J-K, 412, 419, 423.
    por cambios de nivel, 410.
    por flancos, 420.
    por impulsos, 411.
    por niveles, 406.
    principal-subordinado, 411.
    R-S, 406, 412, 418.
    símbolos gráficos, 427.
Binario, sistema, 2.
Bipolar:
  circuito integrado digital, 267.
  transistor, 238.
Biquinario, código, 15.
Bloque, circuito, 243.
Bloque funcional:
  combinacional, 89.
  secuencial síncrono, 455.
Boole, álgebra de, 21.
BPLD, 255.
Bus, 512, 633.

CAF, 355.
Canal (MOS), 291.
Canónico:
  producto, 29.
  suma, 29.

término, 29.
Capa fina, circuito integrado híbrido, 266.
Capa gruesa, circuito integrado híbrido, 266.
Capacidad de una memoria, 578.
Cargabilidad de entrada, 233.
Cargabilidad de salida, 233.
Célula estándar (standard cell), 265.
Ciclo de escritura, 583.
Circuito bloque, 243.
Circuito integrado digital:
híbrido, 245, 266.
monolítico (CIDM), 245:
ADLP, 168.
APLD, 168.
aplicación específica, 250, 261.
arquitectura configurable, 249, 251.
arquitectura fija, 249, 250.
arseniuro de galio, 311.
ASIC, 250, 261.
BDLP, 255.
BICMOS, 309.
bipolar, 267.
BPLD, 255.
CDLP, 255.
célula estándar, 265.
configurable, 251.
conjunto configurable de puertas (FPGA), 256, 258.
conjunto programable por máscara de puertas (MPGA), 264.
CMOS, 303.
CPLD, 255.
DLP, 168, 255.
especificado por el usuario, 245, 261.
estándar, 245, 249.
FPGA, 256, 258.
GLSI, 247.
LSI, 247.
MOS, 290.
MPGA, 264.
MSI, 247.
normalizado, 245, 249.
PLD, 168, 255.
semimedida (semicustom), 264.
SSI, 247.
standard off the shelf, 245, 249.
totalmente a medida (full custom), 264.
ULSI, 247.
VLSI, 247.
CMOS, 304.
Codificación de estados internos:
sistemas secuenciales asíncronos, 324, 357.
sistemas secuenciales síncronos:
con registros de entrada y salida en paralelo, 518, 521.
con contadores síncronos, 526.
Codificador, 108:
con prioridad, 109.
sin prioridad, 109.
Codificador de posición absoluto óptico, 11.
Códigos:
Aiken, 8.
alfanuméricos, 13.
autocomplementarios, 9.
binarios, 7.
biquinario, 14.
cíclicos, 10.
continuos, 10.
correctores de errores, 17.
decimales codificados en binario (BCD), 8.
detectores de errores, 14.
dos entre cinco, 14.
exceso tres, 9.
Gray, 10.
Hamming, 18.
Johnson, 12.
ponderados, 8.
COLA, memoria, 649.
Coma fija, formato de, 227.
Coma flotante, formato de, 227.
Combinacional, sistema, 39.
codificador, 108.
comparador, 134.
decodificador, 90.
decodificador programable (PGA), 141.
detector de identidad, 141.
demultiplexor, 106.
detector/generador de paridad, 136.
multiplexor, 112.
multiplexor/demultiplexor, 126.
programable, 139:
no universal, 140.
universal completo, 144.
universal incompleto, 154.
Comparación binaria, 36.
Comparador digital, 134.
Complemento:
a dos, 199.
a nueve, 219.
a uno, 204.
Configurable, CIDM, 251:
organización cuadrícula, 259.
organización mar de puertas, 260.
organización matricial, 254.
organización terraza, 258.
Conjunto configurable de puertas (FPGA), 258.
Conjunto programable por máscara de puertas (MPGA), 264.
Conmutación:
álgebra de, 22.
diodo en, 236.
MOS en, 296.
transistor bipolar en, 238.
Contadores:
acoplamiento de, 473, 483.
asíncronos, 451, 471.
asíncrono BCD natural, 473.
asíncrono binario natural, 471.

código de contaje, 470.
de capacidad variable, 479, 486.
modo de operación, 470.
síncronos, 441, 481:
con entrada de inhibición, 443, 481.
con entradas en paralelo síncronas, 443, 481.
reversible, 442.
visualización de, 490.
Control, unidad de, 562.
Controlador lógico síncrono, 566.
Conversión de números:
enteros, 2.
fraccionarios, 4.
Convertidor de código, 103.
Corrección de errores, 17.
CPLD, 255.
Cronograma, 571.
Curva de transferencia, 234.

D cerrojo (latch), biestable, 407.
D de activación por flancos (edge-triggered), biestable, 421.
D principal-subordinado, 417.
De Morgan, leyes de, 25.
Decodificador, 90:
excitador, 92, 102.
no excitador, 92.
programable (PGA), 141.
Demultiplexor, 106.
Depletion (empobrecido) (MOS), 292.
Desinhibición, entrada de, 90, 458.
Desplazamiento:
registro de, 447, 496, 640.
Detección de errores, 14.
Detector/generador de paridad, 136.
Detector de identidad, 141.
Diagrama:
de bloques de un procesador digital secuencial, 562.
de flujo de un sistema secuencial asíncrono, 325, 431.
de flujo de un sistema secuencial síncrono, 434.
Dinámica, memoria de acceso aleatorio, 595, 674.
Diodo en conmutación, 236.
Dirección, variable de, 578.
Direccionamiento, 578.
Dispositivo lógico programable, 168, 255, 559.
Distancia, 14.
Divisor de frecuencia, 496.
DLP, 168, 255, 559:
avanzado (ADLP), 255.
básico (BDLP), 255.
complejo (CDLP), 255.
Drenador (MOS), 290.
DTL, 267.
Dualidad, principio de, 22.
ECL, 283.
EEPROM, memoria, 620.
Empobrecido (depletion) (MOS), 292.
Enhancement (enriquecido) (MOS), 292.
EEPROM, memoria pasiva, 610.
EEPROM serie, 626.
Equivalencia, función, 36.
Errores en códigos binarios:
corrección, 17.
detección, 14.
Escalas de integración:
giga grande, 247.
grande, 247.
media, 247.
muy grande, 247.
pequeña, 247.
ultra grande, 247.
Escritura de una memoria 583.
Esencial, término primo, 52.
Estados:
equivalentes, 327.
pseudoequivalentes, 327.
Estados internos, 315:
de transición, 344.
estables, 323.
inestables, 323.
Estática, memoria de acceso aleatorio, 592, 667.
Exceso tres, código, 9.
Excitación, matriz de, 322.
Expansibilidad, 250.
Expresión canónica, 30.

Fases, tabla de, 326.
FIFO, 649.
Flanco, biestable activado por, 354, 420.
FLASH, memoria, 629.
Flujo:
diagrama de un sistema secuencial asíncrono, 325, 431.
diagrama de un sistema secuencial síncrono, 432.
tabla de, 324.
Formato, representación números fraccionarios:
coma fija, 227.
coma flotante, 227.
FPGA, 256, 258.
Fraccionarios, números:
coma fija, 227.
coma flotante, 228.
Función:
canónica, 30.
de un álgebra de Boole, 29.
equivalencia, 36.
incompleta, 54.
inversión, 21.
NO-O, 26.
NO-Y, 26.
O, 24, 26.
O-exclusiva, 32.

Y, 24, 26.
Funcionamiento, tabla de, 323.

Generador de acarreos, 185.
Generador de impulsos, 385, 394.
GLSI, 247.
Graduador (MOS), 290.
Grafo de transición, 346.
Gray, código, 10.

Hamming, código de, 18.
Hexadecimal, sistema de numeración, 6.
HTL, 279.

Impulsos, generador de, 385, 394.
Incompleta, función, 54.
Inhibición, entrada de, 90, 458.
Inmunidad dinámica al ruido, 235.
Integrado, circuito digital:
arseniuro de galio, 311.
BICMOS, 309.
bipolar, 245, 267.
bipolar y MOS (BICMOS), 309.
hibrido, 245, 266.
monolítico, 245.
MOS, 245, 290.
Interno, estado, 315.
Interruptores analógicos, 124.
Interruptores electrónicos, 124, 305.
Inversión, lógica, 21.

J-K, biestable asincrono, 332.
J-K, biestable síncrono:
de activación por flancos, 424.
principal-subordinado, 412.
Johnson, código, 12.

Karnaugh, tabla de, 42.

Lectura, de una memoria, 579, 582.
LIFO, 655.
Lógica:
de acoplamiento por emisor, (ECL), 283.
de alta inmunidad al ruido, (HTL), 279.
de inyección integrada (IIL), 285.
diodo-transistor (DTL), 267.
negativa, 233.
no saturada, 281.
positiva, 233.
resistencia-transistor (RTL), 242, 243, 267.
transistor-transistor (TTL), 272.
transistor-transistor de tres estados, 276.
transistor-transistor Schottky, 281.
LSI, 247.

Margen de ruidos, 234.
Matriz de excitación, 322.
Matriz de salida, 322.
Matriz lógica programable (PLA), 155.
Matriz lógica programable de puertas universales, 174.
Matriz lógica Y-programable (PAL), 161.
McCluskey, método numérico de simplificación de, 49.
Mealy, autómata de, 316.
Memoria:
acceso aleatorio, 581.
acceso múltiple, 604.
activa, 591.
estática, 592
dinámica, 592.
escritura y lectura no simultáneas, 593.
escritura y lectura simultáneas, 600.
pasivas programables, 609, 676.
pasivas reprogramables, 609, 680.
totalmente pasivas, 608, 676, 680.
acceso serie, 636.
PILA (LIFO), 655.
TUBO (FIFO), 649.
registro de desplazamiento, 640.
asociativa, 659.
Memoria, unidad de, 577.
Memorias temporal (buffer), 510.
Microcomputador, 596, 619.
Microcontrolador, 626.
Microinstrucción, 538.
Microprocesador, 596, 620.
Microprogramable,
sistema secuencial síncrono, 517, 530.
Modularidad de sistemas secuenciales síncronos:
de entradas, 552.
de salidas, 553.
Monoestable, 371.
no realimentado, 373.
no redisparable, 380.
realimentado, 374.
redisparable, 381.
Monolítico, circuito integrado, 245.
Moore, autómata de, 316.
MOS:
dinámico, 301.
estático, 299.
MOS en conmutación, 296.
MPGA, 264.
MSI, 247.
Multifunción, 60.
Multiplexor analógico, 126.
Multiplexor digital, 112:
realizado con interruptores, 124.
realizado con puertas Y y O, 112.
realizado con puertas de tres estados, 121.
Multiplexor/demultiplexor, 126.
Multiplicación binaria, 211.

NAND (NO-Y), función, 26.
Negativos, números, 206.
Niveles,
biestables asíncronos activados por, 331.

sistemas secuenciales asíncronos, 321.
NO-O (NOR), función, 26.
NOR (NO-O), función, 26.
NO-Y (NAND), función, 26.
Numeración, sistemas de, 1.
Numérico, método de simplificación, 49.
Números binarios, 2.
Números fraccionarios, 225.

O, operación, 24, 26.
Octal, sistema, 5.
O-exclusiva:
función, 32.
puerta, 32.
Operación:
inversión, 21.
O, 24, 26.
Y, 24, 26.
Operativa, unidad, 562.
Organización de una memoria de acceso aleatorio:
2D, 585.
2 ½ D, 589.
3D, 587.
Organización serie:
bit a bit, 638.
posición a posición, 639.
Osciladores de onda cuadrada, 385, 394.

PAL, 161.
Paridad:
bit de, 16.
detector, 136.
generador de, 136.
Pasivas, memorias:
programables, 609, 676.
reprogramables, 609, 680.
totalmente, 608, 676, 680.
Periférico, 766, 825, 831.
Permanencia de la información en una memoria, 591.
PILA (LIFO), 655.
PLA, 155.
PLD, 168, 255, 559.
Ponderado, código, 8.
Posición de memoria, 578.
Potencia disipada, 236.
PQ, biestable, 333.
Primo, término, 50.
Procesador de comunicaciones, 626, 654.
Procesador digital:
combinacional, 559.
secuencial, 561:
asíncrono, 563.
síncrono, 562.
Producto:
canónico, 29.
de sumas, 29.
lógico, 24, 26.
Producto potencia-tiempo de propagación, 236.
Programables, circuitos combinacionales, 139.
Programables, memorias pasivas, 609, 676.
Programador de memorias, 613.
PROM, 609, 676.
Puerta (MOS), 290.
Puerta lógica:
NO-O, 26.
NO-Y, 26.
O-exclusiva, 34, 79.

Quine, método numérico de, 49.

RAM, 144, 593.
Reacción directa, 318.
Realimentación:
con células asíncronas, 318.
células o biestables asíncronos activados por flancos, 354.
células o biestables asíncronos activados por niveles, 331.
directa, 318.
Rebasamiento, 202.
Registro:
conjunto de, 510.
de entrada y salida en paralelo, 437, 509.
de desplazamiento:
estático, 447, 496, 640.
entradas en paralelo, 498.
reversible, 504.
dinámico, 641, 683.
Reprogramable, memoria pasiva, 609, 680.
Resta:
BCD natural, 219.
binario natural, 198, 199, 204.
Restador BCD natural, 219.
Restador binario natural:
complemento a dos, 199.
complemento a uno, 204.
ROM, 148, 608, 676, 680.
RPROM, 609, 680.
R-S, biestable, 330.
RTL, 242, 243, 267.
Ruido:
inmunidad dinámica, 235.
margen de, 234.

Salida de tres estados, 276.
Salida, matriz de, 322.
Saturación del transistor, 238.
Schottky, 281.
Secuencia de operaciones, diagrama de:
de un sistema secuencial asíncrono, 346.
Secuencial, sistema:
asíncrono, 318, 320.
síncrono, 320, 402.
Semimedida (Semicustom), circuito integrado, 264.
Semisumador, 180.

Serial EEPROM, 626.
Símbolos lógicos normalizados, 687:
  asociados a entradas y salidas, 691.
  biestables, 427.
  bloques funcionales síncronos, 464.
  relaciones de dependencia:
    combinación de, 704, 712.
    conexión, 703.
    control, 709.
    desinhibición/inhibición, 701.
    direccionamiento, 703.
    inversión, 700.
    modo de operación, 702, 712.
    O, 700.
    Y, 698.
  sistemas combinacionales, 695.
  sistemas secuenciales, 707.
  Simplificación de las funciones lógicas, 40.
  Síncrono, sistema secuencial, 320, 402, 516:
    asincronizado, 448.
    cableado, 516.
    con contadores síncronos, 522.
    microprogramable, 517, 530.
    modular, 517, 552.
    no modular, 517.
    semimodular, 517, 556.
Sistema de numeración:
  binario, 2.
  decimal, 2.
  hexadecimal, 6.
  octal, 5.
SSI, 247.
Suma:
  aritmética BCD, 216.
  aritmética binaria, 179.
  canónica, 29.
  de productos, 31.
  lógica, 24, 26.
Sumador digital:
  BCD natural, 216.
  binario natural, 179.
  con acarreo paralelo, 184.
  con acarreo serie, 183.
  paralelo, 183.
  serie, 183.
  total, 181, 191.
Sumador restador:
  BCD natural, 225.
  binario natural, 202, 208.
Surtidor (MOS), 290.

T, biestable, 333.
Tabla:
  de fases, 326.
  de flujo, 324.
  de funcionamiento, 323.
  de Karnaugh, 42.
  de Veitch, 42.
  de verdad, 33.
Tecnología:
  CMOS, 303.
  de acoplamiento por emisor (ECL), 283.
  de alta inmunidad al ruido (HTL), 279.
  de inyección integrada (IIL), 285.
  diodo-transistor (DTL), 267.
  MOS dinámico, 301.
  MOS estático, 299.
  Schottky, 281.
  transistor-transistor (TTL), 272.
Temporales, circuitos digitales, 371.
Temporizadores:
  analógico-digitales, 383, 394.
  digitales, 371.
Tensión de umbral, 234.
Teorema:
  de absorción, 24.
  de De Morgan, 25.
Término:
  canónico, 29.
  primo, 50.
  primo esencial, 52.
Tiempo:
  de acceso, 582.
  de almacenamiento, 241.
  de ciclo, 583.
  de propagación medio, 235.
Totalmente a medida, circuito integrado, 264.
Transferencia:
  curva de, 234.
Transiciones no adyacentes:
  críticas, 324.
  no críticas, 324.
Transistor bipolar en conmutación, 238.
Tres estados, salida de, 276.
TTL, 271.
TUBO (FIFO), 649:
  con memoria de acceso aleatorio, 649, 652.
  con registro de desplazamiento, 649.

ULSI, 247.
Umbral, tensión de, 234.
Unidad:
  aritmética y lógica, 208.
  de control, 562.
  de memoria, 577.
  operativa, 562.
USIC, 261.

Vitch, tabla de, 42.
Verdad, tabla de, 33.
Visualizador (display):
  estático, 493.
  dinámico, 494.
VLSI, 247.
Volatilidad, 592, 676.

Y, operación, 24, 26.
Y por conexión, 77.

Esta nueva edición de Sistemas Electrónicos Digitales incorpora los avances producidos en la Electrónica Digital durante la década de los ochenta y de los noventa. El espectacular desarrollo de la microelectrónica ha intensificado la tendencia generalizada, ya iniciada en la década de los setenta, a aumentar la complejidad del sistema físico (hardware) para elevar la velocidad de los procesadores digitales y ampliar de esa forma su campo de aplicación. Un ejemplo de esto es el desarrollo de los circuitos digitales configurables. Esto hace que el ingeniero se vea obligado a cambiar sus métodos de diseño y a elevar su capacidad de síntesis de estructuras digitales complejas.

El desarrollo de las técnicas hipermedia que está llamado a revolucionar las metodologías educativas se utiliza en esta edición para proporcionar al lector la posibilidad de autoevaluarse y de estudiar los conceptos básicos de circuitos integrados digitales de forma interactiva.

EL AUTOR:

Enrique Mandado Pérez, Perito Industrial Eléctrico por la Escuela de Peritos Industriales de Vigo con el número uno de su promoción (1963). Ingeniero Industrial por la Universidad Politécnica de Madrid con Premio Nacional de Fin de carrera (1969). Doctor Ingeniero Industrial por la Universidad Politécnica de Cataluña (1976) en la que fue profesor de Electrónica y Computadores desde 1971 a 1978. Durante 10 años simultaneó la actividad docente con su trabajo como ingeniero de diseño de sistemas electrónicos de control en los laboratorios de COPRESA (Miniwatt) y como consultor de diversas empresas. En 1978 recibió el Premio Mundo Electrónico al Mejor Libro de Electrónica. En 1996 recibió el Premio Xunta de Galicia al Mejor Trabajo del Ámbito Tecnológico. Actualmente es Catedrático y Director del Instituto de Electrónica Aplicada Pedro Barrié de la Maza de la Universidad de Vigo. Sus principales campos de investigación y desarrollo son la innovación tecnológica mediante la implantación de equipos electrónicos y, en particular, el diseño asistido por computador de circuitos integrados y las comunicaciones digitales.

ISBN: 84-267-1170-7

9 788426 711700